Hochschule Zittau/Görlitz
UNIVERSITY OF APPLIED SCIENCES

Faculty of
MECHANICAL ENGINEERING
Department of
TECHNICAL THERMODYNAMICS

Property Library for Mixtures of Water/Lithium Bromide

FluidLAB with LibWaLi for MATLAB®

Prof. Hans-Joachim Kretzschmar
Dr. Ines Stoecker
Matthias Kunick
S. Hasch

Property Software for Water/Lithium Bromide

FluidLAB LibWaLi for MATLAB®

Contents

© Zittau/Goerlitz University of Applied Sciences, Germany
Faculty of Mechanical Engineering
Department of Technical Thermodynamics
Prof. Hans-Joachim Kretzschmar
Dr. Ines Stoecker
Phone: +49-3583-61-1846 or -1881
Fax: +49-3583-61-1846
E-mail: hj.kretzschmar@hs-zigr.de
Internet: www.thermodynamics-zittau.de

0 Package Contents

0.1 Zip file for 32-bit MATLAB®

The following zip file is delivered for your computer running a 32-bit version of MATLAB®.

"CD_FluidLAB_LibWaLi.zip"

Including the following files:

FluidLAB_LibWaLi_Setup.exe	- Installation program for the FluidLAB Add-On for use in MATLAB®
LibWaLi.dll	- Dynamic Link Library for Water/Lithium Bromide for use in MATLAB®
FluidLAB_LibWaLi_Docu_Eng.pdf	- User's Guide

0.2 Zip file for 64-bit MATLAB®

The following zip file is delivered for your computer running a 64-bit version of MATLAB®.

"CD_FluidLAB_LibWaLi_x64.zip"

Including the following files and folders:

Files:

Setup.exe	- Self-extracting and self-installing program for FluidLAB
FluidLAB_LibWaLi_64.msi	- Installation program for the FluidLAB Add-On for use in MATLAB®
LibWaLi.dll	- Dynamic Link Library for Water/Lithium Bromide for use in MATLAB®
FluidLAB_LibWaLi_Docu_Eng.pdf	- User's Guide

Folders:

vcredist_x64	- Folder containing the "Microsoft Visual C++ 2010 x64 Redistributable Pack"
WindowsInstaller3_1	- Folder containing the "Microsoft Windows Installer"

1. Property Functions

Functional Dependence	Function Name	Call from Fortran program	Call in DLL LibWaLi as parameter	Property or Function	Unit of the result
$c_p = \mathrm{f}(p,t,\xi)$	cp_ptxi_WaLi	CP_PTXI_WALI(P,T,XI)	C_CP_PTXI_WALI(CP,P,T,XI)	Specific isobaric heat capacity	kJ/(kg K)
$c'_p = \mathrm{f}(p_\mathrm{s},t_\mathrm{s},\xi')$	cpl_pstsxil_WaLi	CPL_WALI(PS,TS,XIL)	C_CPL_WALI(CPL,PS,TS,XIL)	Specific isobaric heat capacity of saturated liquid	kJ/(kg K)
$c''_p = \mathrm{f}(p_\mathrm{s},t_\mathrm{s},\xi')$	cpv_pstsxil_WaLi	CPV_WALI(PS,TS,XIL)	C_CPV_WALI(CPV,PS,TS,XIL)	Specific isobaric heat capacity of saturated steam	kJ/(kg K)
$\eta = \mathrm{f}(p,t,\xi)$	eta_ptxi_WaLi	ETA_PTXI_WALI(P,T,XI)	C_ETA_PTXI_WALI(ETA,P,T,XI)	Dynamic viscosity	Pa s
$\eta' = \mathrm{f}(p_\mathrm{s},t_\mathrm{s},\xi')$	etal_pstsxil_WaLi	ETAL_WALI(PS,TS,XIL)	C_ETAL_WALI(ETAL,PS,TS,XIL)	Dynamic viscosity of saturated liquid	Pa s
$\eta'' = \mathrm{f}(p_\mathrm{s},t_\mathrm{s},\xi')$	etav_pstsxil_WaLi	ETAV_WALI(PS,TS,XIL)	C_ETAV_WALI(ETAV,PS,TS,XIL)	Dynamic viscosity of saturated steam	Pa s
$h = \mathrm{f}(p,t,\xi)$	h_ptxi_WaLi	H_PTXI_WALI(P,T,XI)	C_H_PTXI_WALI(H,P,T,XI)	Specific enthalpy	kJ/kg
$h' = \mathrm{f}(p_\mathrm{s},t_\mathrm{s},\xi')$	hl_pstsxil_WaLi	HL_WALI(PS,TS,XIL)	C_HL_WALI(HL,PS,TS,XIL)	Specific enthalpy of saturated liquid	kJ/kg
$h'' = \mathrm{f}(p_\mathrm{s},t_\mathrm{s},\xi')$	hv_pstsxil_WaLi	HV_WALI(PS,TS,XIL)	C_HV_WALI(HV,PS,TS,XIL)	Specific enthalpy of saturated steam	kJ/kg
$h_\mathrm{sol} = \mathrm{f}(\xi)$	hsol_xi_WaLi	HSOL_XI_WALI(XI)	C_HSOL_XI_WALI(HSOL,XI)	Specific enthalpy at crystallization barrier	kJ/kg
$\lambda = \mathrm{f}(p,t,\xi)$	lam_ptxi_WaLi	LAM_PTXI_WALI(P,T,XI)	C_LAM_PTXI_WALI(LAM,P,T,XI)	Thermal conductivity	W/(m K)
$\lambda' = \mathrm{f}(p_\mathrm{s},t_\mathrm{s},\xi')$	laml_pstsxil_WaLi	LAML_WALI(PS,TS,XIL)	C_LAML_WALI(LAML,PS,TS,XIL)	Thermal conductivity of saturated liquid	W/(m K)
$\lambda'' = \mathrm{f}(p_\mathrm{s},t_\mathrm{s},\xi')$	lamv_pstsxil_WaLi	LAMV_WALI(PS,TS,XIL)	C_LAMV_WALI(LAMV,PS,TS,XIL)	Thermal conductivity of saturated steam	W/(m K)
$\nu = \mathrm{f}(p,t,\xi)$	ny_ptxi_WaLi	NY_PTXI_WALI(P,T,XI)	C_NY_PTXI_WALI(NY,P,T,XI)	Kinematic viscosity	m^2/s
$\nu' = \mathrm{f}(p_\mathrm{s},t_\mathrm{s},\xi')$	nyl_pstsxil_WaLi	NYL_WALI(PS,TS,XIL)	C_NYL_WALI(NYL,PS,TS,XIL)	Kinematic viscosity of saturated liquid	m^2/s
$\nu'' = \mathrm{f}(p_\mathrm{s},t_\mathrm{s},\xi')$	nyv_pstsxil_WaLi	NYV_WALI(PS,TS,XIL)	C_NYV_WALI(NYV,PS,TS,XIL)	Kinematic viscosity of saturated steam	m^2/s
$Pr = \mathrm{f}(p,t,\xi)$	Pr_ptxi_WaLi	PR_PTXI_WALI(P,T,XI)	C_PR_PTXI_WALI(PR,P,T,XI)	*Prandtl*-Number	-

Functional Dependence	Function Name	Call from Fortran program	Call in DLL LibWaLi as parameter	Property or Function	Unit of the result
$Pr' = \mathrm{f}(p_\mathrm{s}, t_\mathrm{s}, \xi')$	Prl_pstsxil_WaLi	PRL_WALI(PS,TS,XIL)	C_PRL_WALI(PRL,PS,TS,XIL)	*Prandtl*-Number of saturated liquid	-
$Pr'' = \mathrm{f}(p_\mathrm{s}, t_\mathrm{s}, \xi')$	Prv_pstsxil_WaLi	PRV_WALI(PS,TS,XIL)	C_PRV_WALI(PRV,PS,TS,XIL)	*Prandtl*-Number of saturated steam	-
$p_\mathrm{s} = \mathrm{f}(t_\mathrm{s}, \xi')$	ps_tsxil_WaLi	PS_TSXIL_WALI(TS,XIL)	C_PS_TSXIL_WALI(PS,TS,XIL)	Vapor pressure	bar
$p_\mathrm{sol} = \mathrm{f}(t)$	psol_t_WaLi	PSOL_T_WALI(T)	C_PSOL_T_WALI(PSOL,T)	Pressure at crystallization barrier	bar
$Region = \mathrm{f}(p, t, \xi)$	region_ptxi_WaLi	REGION_PTXI_WALI(P,T,XI)	C_REGION_PTXI_WALI(REGION,P,T,XI)	Phase region from pressure, temperature and mass fraction of H_2O	-
$Region = \mathrm{f}(p, h, \xi)$	region_phxi_WaLi	REGION_PHXI_WALI(P,H,XI)	C_REGION_PHXI_WALI(REGION,P,H,XI)	Phase region from pressure, enthalpy and mass fraction of H_2O	-
$Region = \mathrm{f}(p, s, \xi)$	region_psxi_WaLi	REGION_PSXI_WALI(P,S,XI)	C_REGION_PSXI_WALI(REGION,P,S,XI)	Phase region from pressure, entropy and mass fraction of H_2O	-
$s = \mathrm{f}(p, t, \xi)$	s_ptxi_WaLi	S_PTXI_WALI(P,T,XI)	C_S_PTXI_WALI(S,P,T,XI)	Specific entropy	kJ/(kg K)
$s' = \mathrm{f}(p_\mathrm{s}, t_\mathrm{s}, \xi')$	sl_pstsxil_WaLi	SL_WALI(PS,TS,XIL)	C_SL_WALI(SL,PS,TS,XIL)	Specific entropy of saturated liquid	kJ/(kg K)
$s'' = \mathrm{f}(p_\mathrm{s}, t_\mathrm{s}, \xi')$	sv_pstsxil_WaLi	SV_WALI(PS,TS,XIL)	C_SV_WALI(SV,PS,TS,XIL)	Specific entropy of saturated steam	kJ/(kg K)
$t = \mathrm{f}(p, h, \xi)$	t_phxi_WaLi	T_PHXI_WALI(P,H,XI)	C_T_PHXI_WALI(T,P,H,XI)	Backward function: Temperature from pressure, enthalpy and mass fraction of H_2O	°C
$t = \mathrm{f}(p, s, \xi)$	t_psxi_WaLi	T_PSXI_WALI(P,S,XI)	C_T_PSXI_WALI(T,P,S,XI)	Backward function: Temperature from pressure, entropy and mass fraction of H_2O	°C
$t_\mathrm{s} = \mathrm{f}(p_\mathrm{s}, \xi')$	ts_psxil_WaLi	TS_PSXIL_WALI(PS,XIL)	C_TS_PSXIL_WALI(TS,PS,XIL)	Saturation temperature	°C

Functional Dependence	**Function Name**	**Call from Fortran program**	**Call in DLL LibWaLi as parameter**	**Property or Function**	**Unit of the result**
$t_{\text{sol}} = \text{f}(p)$	tsol_p_WaLi	TSOL_P_WALI(P)	C_TSOL_P_WALI(TSOL,P)	Temperature at crystallization barrier	°C
$v = \text{f}(p,t,\xi)$	v_ptxi_WaLi	V_PTXI_WALI(P,T,XI)	C_V_PTXI_WALI(V,P,T,XI)	Specific volume	m^3/kg
$v' = \text{f}(p_\text{s},t_\text{s},\xi')$	vl_pstsxil_WaLi	VL_WALI(PS,TS,XIL)	C_VL_WALI(HL,PS,TS,XIL)	Specific volume of saturated liquid	m^3/kg
$v'' = \text{f}(p_\text{s},t_\text{s},\xi')$	vv_pstsxil_WaLi	VV_WALI(PS,TS,XIL)	C_VV_WALI(HV,PS,TS,XIL)	Specific volume of saturated steam	m^3/kg
$\xi' = \text{f}(p_\text{s},t_\text{s})$	xil_psts_WaLi	XIL_PSTS_WALI(PS,TS)	C_XIL_PSTS_WALI(XIL,PS,TS)	Mass fraction H_2O of saturated liquid	kg/kg
$\xi_{\text{sol}} = \text{f}(p)$	xisol_p_WaLi	XISOL_P_WALI(P)	C_XISOL_P_WALI(XISOL,P)	Mass fraction H_2O at crystallization barrier	kg/kg
$\xi'' = \text{f}(p_\text{s},t_\text{s})$	xiv_psts_WaLi	XIV_PSTS_WALI(PS,TS)	C_XIV_PSTS_WALI(XIV,PS,TS)	Mass fraction H_2O of saturated steam	kg/kg

Units:

t	in °C
p	in bar
ξ	in (kg H_2O)/(kg mixture)

Important hints for the calculation of wet steam

The wet steam region is calculated automatically by the subprograms, which are valid within the entire range of state.

It is necessary to define two parameters for the functions of saturated liquids (...′) and saturated steam (...″).

- either t_s and ξ'
- or p_s and ξ'
- or p_s and t_s

and –1 must be entered as a pro-forma value for the value which is not given. When p_s, t_s, and ξ' are entered, the program will consider p_s, t_s, and ξ' to be appropriate to represent the saturation curve $p_{s.}$ If this is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Reference state

Water:	Triple point for saturated liquid
	h_{H2O} = 0.000611783 kJ/kg and
	$s_{H2O} = 0$
	at p_{tr} = 0.00611657 bar and t_{tr} = 0.01 °C
Mixture of water/lithium bromide:	saturated liquid
	$h_{0.5}$ = −0.0209415 kJ/kg and
	$s_{0.5}$ = −0.0000780433 kJ/(kg K)
	at t = 0 °C and ξ = 0.5 kg H_2O / kg

Note.

If the calculation results in –1000, the values entered represent a state point outside the range of validity of LibWaLi. For further information on each function and its range of validity see Chapter 3. The same information may also be accessed via the online help pages.

2 Application of FluidLAB in MATLAB

The FluidLAB Add-In has been developed to calculate thermodynamic properties in MATLAB® more conveniently. Within MATLAB®, it enables the direct call of functions relating to water/lithium bromide from the LibWaLi property library.

2.1 Installing FluidLAB LibWaLi

Installing FluidLAB including LibWaLi for 32-bit MATLAB®

This section describes the installation of FluidLAB LibWaLi for a 32-bit version of MATLAB®.

Before you begin, it is best to close any Windows® applications, since Windows® may need to be rebooted during the installation process.

After you have downloaded and extracted the zip-file "CD_FluidLAB_LibWaLi.zip", you will see the folder

CD_FluidLAB_LibWaLi

in your Windows Explorer®, Norton Commander® or another similar program you are using.

Open this folder by double-clicking on it.

In this folder you will see the following files:

FluidLAB_LibWaLi_Docu_Eng.pdf
FluidLAB_LibWaLi_Setup.exe
LibWaLi.dll.

In order to run the installation of FluidLAB including, the LibWaLi property library, double-click on the file

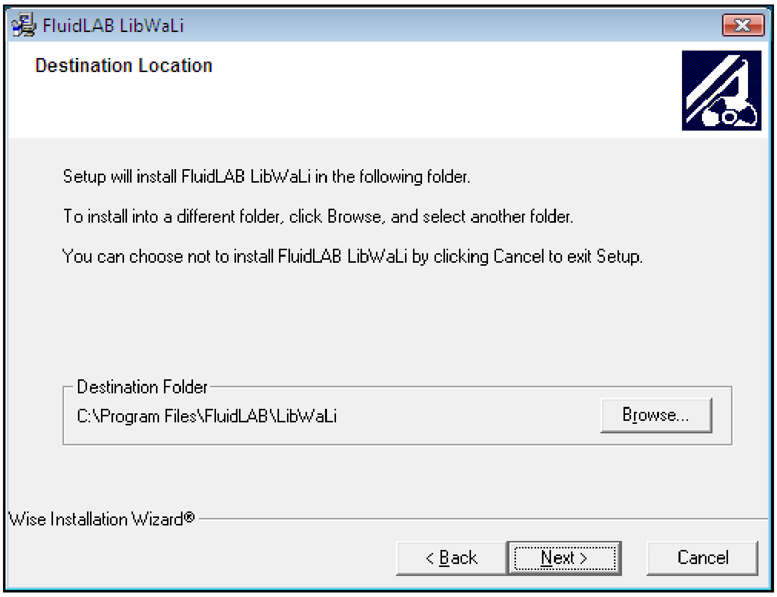
FluidLAB_LibWaLi_Setup.exe.

Installation may start with a window noting that all Windows® programs should be closed. When this is the case, the installation can be continued. Click the "Next >" button.

In the following dialog box, "Destination Location", the default path offered automatically for the installation of FluidLAB is

C:\Program Files\FluidLAB\LibWaLi (for English version of Windows)
C:\Programme\FluidLAB\LibWaLi (for German version of Windows).

By clicking the "Browse…" button, you can change the installation directory before installation (see Figure 2.1).



FluidLAB LibWaLi

Destination Location

Setup will install FluidLAB LibWaLi in the following folder.

To install into a different folder, click Browse, and select another folder.

You can choose not to install FluidLAB LibWaLi by clicking Cancel to exit Setup.

Destination Folder: C:\Program Files\FluidLAB\LibWaLi

Figure 2.1: Dialog window "Destination Location"

If you wish to change directories, click the "Browse…" button and select your desired directory. The instructions in this documentation refer to the stated default directory. Leave this window by clicking the "Next >" button.

The dialog window "Start Installation" pops up. Click the "Next >" button to continue installation. The FluidLAB files are now being copied into the created directory on your hard drive. Click the "Finish >" button in the following window to complete installation.

The installation program has copied the following files for LibWaLi into the directory

"C:\Program Files\FluidLAB\LibWaLi" (for English version of Windows):
"C:\Programme\FluidLAB\LibWaLi" (for German version of Windows):

advapi32.dll
Dformd.dll
Dforrt.dll
INSTALL.LOG
LibWaLi.dll
LC.dll
msvcp60.dll
msvcrt.dll
Unwise.exe
Unwise.ini

Now, you have to overwrite the file "LibWaLi.dll" in your FluidLAB directory with the file of the same name provided in your CD folder with FluidLAB.

To do this, open the CD folder in "My Computer" and click on the file "LibWaLi.dll" in order to highlight it.

Then click on the "Edit" menu in your Explorer and select "Copy".

Now, open your FluidLAB directory (the standard being

C:\Program Files\FluidLAB\LibWaLi (for English version of Windows)
C:\Programme\FluidLAB\LibWaLi (for German version of Windows))

and insert the file "LibWaLi.dll" by clicking the "Edit" menu in your Explorer and then select "Paste". Answer the question whether you want to replace the file by clicking the "Yes" button. Now, you have overwritten the file "LibWaLi.dll" successfully and the property functions are available in MATLAB®.

Installing FluidLAB including LibWaLi for 64-bit MATLAB®

This section describes the installation of FluidLAB LibWaLi.

Before you begin, it is best to close any Windows® applications, since Windows® may need to be rebooted during the installation process.

After you have downloaded and extracted the zip-file "CD_FluidLAB_LibWaLi_x64.zip", you will see the folder

CD_FluidLAB_LibWaLi

in your Windows Explorer®, Norton Commander® or other similar program you are using.

Open this folder by double-clicking on it.

In this folder you will see the following files

FluidLAB_LibWaLi_Docu_Eng.pdf
FluidLAB_LibWaLi_64_Setup.msi
LibWaLi.dll
Setup.exe.

and folders

/vcredist_x64
/WindowsInstaller3_1.

In order to run the installation of FluidLAB including, the LibWaLi property library, double-click on the file

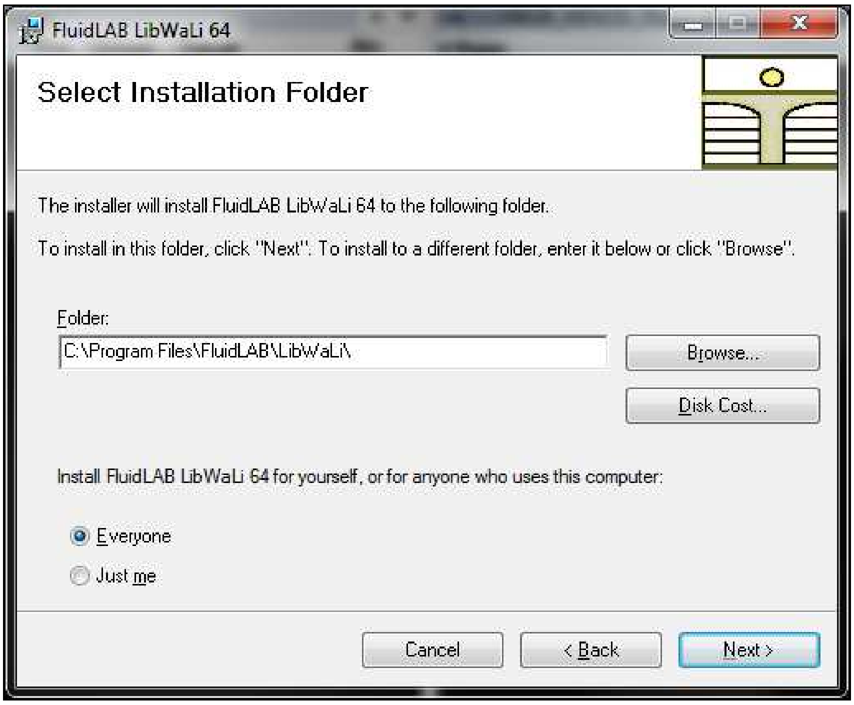
Setup.exe.

Installation of FluidLAB LibWaLi starts with a window noting that the installer will guide you through the installation process. Click the "Next >" button to continue.

In the following dialog box, "Destination Location", the default path offered automatically for the installation of FluidLAB is

C:\Program Files\FluidLAB\LibWaLi (for English version of Windows)
C:\Programme\FluidLAB\LibWaLi (for German version of Windows)

By clicking the "Browse…" button, you can change the installation directory before installation (see Figure 2.2).



FluidLAB LibWaLi 64

Select Installation Folder

The installer will install FluidLAB LibWaLi 64 to the following folder.

To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".

Folder: C:\Program Files\FluidLAB\LibWaLi\

Install FluidLAB LibWaLi 64 for yourself, or for anyone who uses this computer:

● Everyone

○ Just me

Figure 2.2: "Select Installation Folder"

Finally, click on "Next >" to continue installation; click "Next >" again in the "Confirm Installation" window which follows in order to start the installation of FluidLAB.

After FluidLAB has been installed, you will see the sentence "FluidLAB LibWaLi 64 has been successfully installed." Confirm this by clicking the "Close" button.

The installation program has copied the following files for LibWaLi into the directory

"C:\Program Files\FluidLAB\LibWaLi" (for English version of Windows)
"C:\Programme\FluidLAB\LibWaLi" (for German version of Windows):

capt_ico_big.ico
LC.dll
LibWaLi.dll
libifcoremd.dll
libiomp5md.dll
libmmd.dll

Now, you have to overwrite the file "LibWaLi.dll" in your FluidLAB directory with the file of the same name provided in your CD folder with FluidLAB.
To do this, open the CD folder in "My Computer" and click on the file "LibWaLi.dll" in order to highlight it.
Then click on the "Edit" menu in your Explorer and select "Copy".
Now, open your FluidLAB directory (the standard being

C:\Program Files\FluidLAB\LibWaLi (for English version of Windows)
C:\Programme\FluidLAB\LibWaLi (for German version of Windows))

and insert the file "LibWaLi.dll" by clicking the "Edit" menu in your Explorer and then select "Paste". Answer the question whether you want to replace the file by clicking the "Yes" button. Now, you have overwritten the file "LibWaLi.dll" successfully and the property functions are available in MATLAB.

The installation programs for both the 32-bit and the 64-bit Windows version have copied the following function files for LibWaLi into the directory

"C:\Program Files\FluidLAB\LibWaLi" (for English version of Windows)
"C:\Programme\FluidLAB\LibWaLi" (for German version of Windows):

- MATLAB®-Interface-Program for calculable functions

cp_ptxi_WaLi
cpl_pstsxil_WaLi
cpv_pstsxil_WaLi
eta_ptxi_WaLi
etal_pstsxil_WaLi
etav_pstsxil_WaLi
h_ptxi_WaLi
hl_pstsxil_WaLi
hv_pstsxil_WaLi
hsol_xi_WaLi
lam_ptxi_WaLi
laml_pstsxil_WaLi
lamv_pstsxil_WaLi
ny_ptxi_WaLi
nyl_pstsxil_WaLi
nyv_pstsxil_WaLi
Pr_ptxi_WaLi
Prl_pstsxil_WaLi
Prv_pstsxil_WaLi
ps_tsxil_WaLi
psol_t_WaLi
region_ptxi_WaLi
region_phxi_WaLi
region_psxi_WaLi
s_ptxi_WaLi
sl_pstsxil_WaLi
sv_pstsxil_WaLi
t_phxi_WaLi
t_psxi_WaLi
ts_psxil_WaLi
tsol_p_WaLi
v_ptxi_WaLi
vl_pstsxil_WaLi
vv_pstsxil_WaLi
xil_psts_WaLi
xiv_psts_WaLi
xisol_p_WaLi
xisol_t_WaLi

Please note that there is a difference in the file extension of the function files.

The 32-bit installation program has copied function files with the file extension

.mexw32

and the 64-bit installation program has copied function files with the file extension

.mexw64

into your LibWaLi directory (the standard being

C:\Program Files\FluidLAB\LibWaLi (for English version of Windows)
C:\Programme\FluidLAB\LibWaLi (for German version of Windows).

Now, you have to overwrite the file "LibWaLi.dll" in your LibWaLi directory with the file of the same name provided in your CD folder with FluidLAB.
To do this, open the CD folder in "My Computer" and click on the file "LibWaLi.dll" in order to highlight it.

Then click on the "Edit" menu in your Explorer and select "Copy".

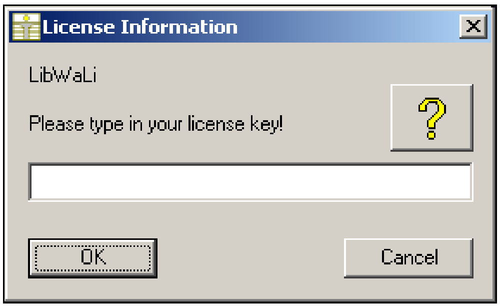
Now, open your FluidLAB directory (the standard being

C:\Program Files\FluidLAB\LibWaLi (for English version of Windows)
C:\Programme\FluidLAB\LibWaLi (for German version of Windows))

and insert the file "LibWaLi.dll" by clicking the "Edit" menu in your Explorer and then select "Paste".
Answer the question whether you want to replace the file by clicking the "Yes" button. Now, you have overwritten the file "LibWaLi.dll" successfully.

2.2 Licensing the LibWaLi Property Library

The licensing procedure has to be carried out when a FluidLAB prompt message appears in MATLAB®. In this case, you will see the "License Information" window for LibWaLi (see figure below).

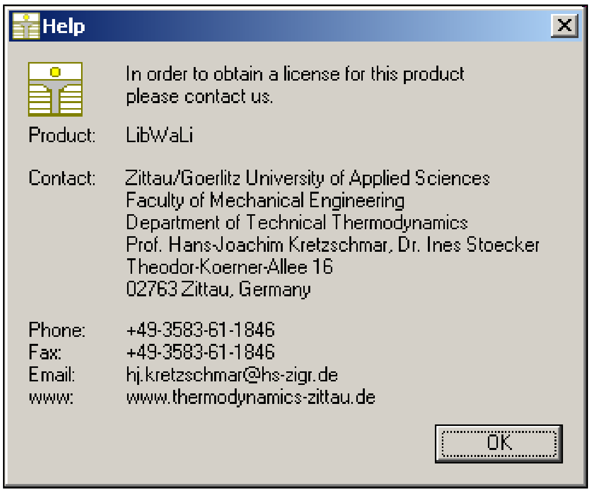


LibWaLi

Please type in your license key!

Figure 2.3: "License Information" window

Here you will have to type in the license key which you have obtained from the Zittau/Goerlitz University of Applied Sciences. You can find contact information on the "Content" page of this User's Guide or by clicking the yellow question mark in the "License Information" window. Then the following window will appear:



In order to obtain a license for this product
please contact us.

Product: LibWaLi

Contact: Zittau/Goerlitz University of Applied Sciences
Faculty of Mechanical Engineering
Department of Technical Thermodynamics
Prof. Hans-Joachim Kretzschmar, Dr. Ines Stoecker
Theodor-Koerner-Allee 16
02763 Zittau, Germany

Phone: +49-3583-61-1846
Fax: +49-3583-61-1846
Email: hj.kretzschmar@hs-zigr.de
www: www.thermodynamics-zittau.de

Figure 2.4: "Help" window

If you do not enter a valid license it is still possible to use MATLAB® by clicking "Cancel". In this case, the LibWaLi property library will display the result "–11111111" for every calculation.
The "License Information" window will appear every time you use FluidLAB LibWaLi unless you uninstall FluidLAB according to the description in section 2.5 of this User's Guide.
Should you not wish to license the LibWaLi property library, you have to delete the file

LibWaLi.dll

in the installation folder of FluidLAB (the standard being

C:\Program Files\FluidLAB\LibWaLi)

using an appropriate program such as Windows Explorer® or Norton Commander®.

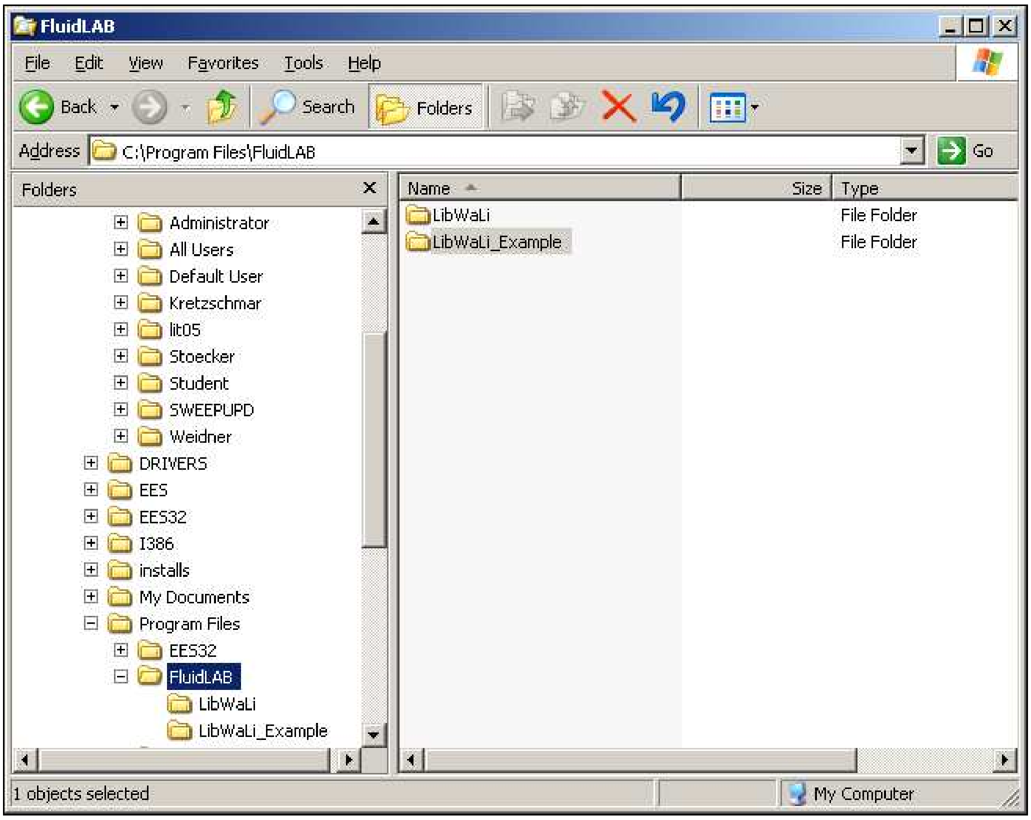
2.3 Example: Calculation of the Specific Enthalpy h = f(p,t,ξ) for Water/Lithium Bromide in an M-File

Now we will calculate, step by step, the specific enthalpy h as a function of pressure p, temperature t and mass fraction ξ for water/lithium bromide using FluidLAB.
Please carry out the following instructions:

- Start Windows Explorer®, Total Commander®, My Computer® or another file manager program.
  The following description refers to Windows Explorer®
- Your Windows Explorer® should be set to "Details" for easier viewing. Click the "Views" button and select "Details."
- Switch into the program directory of FluidLAB in which you will find the folder "\LibWaLi"; it is generally saved under:

  "C:\Program Files\FluidLAB" (for English version of Windows)
  "C:\Programme\FluidLAB" (for German version of Windows)

- Create the folder "\LibWaLi_Example" by clicking on "File" in the Explorer® menu, then "New" in the menu which appears and afterwards selecting "Folder". Name the new folder "\LibWaLi_Example ".
- You will now see the following window:



Address C:\Program Files\FluidLAB

Name	Size	Type
LibWaLi		File Folder
LibWaLi_Example		File Folder

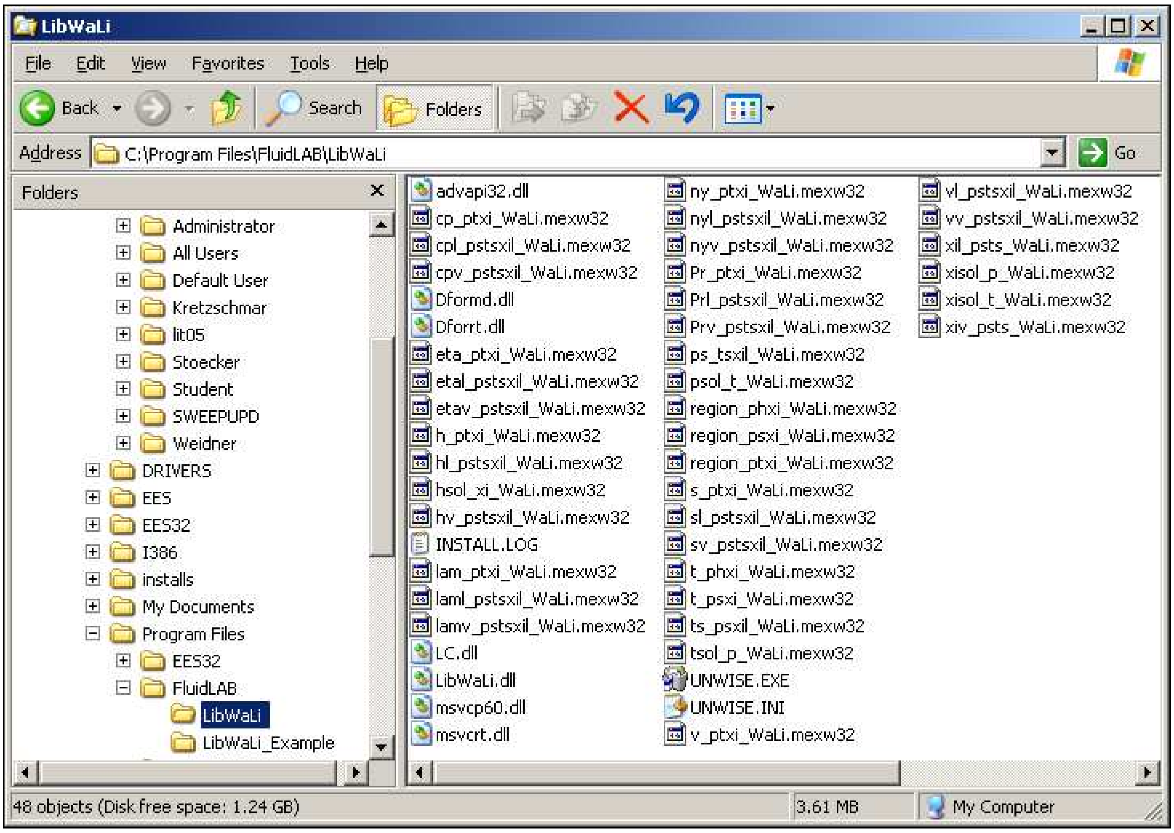
1 objects selected

Figure 2.5: Folder "FluidLAB"

- Switch into the directory "\LibWaLi" within "\FluidLAB", in the standard being

  "C:\Program Files\FluidLAB\LibWaLi" (for English version of Windows)
  "C:\Programme\FluidLAB\LibWaLi" (for German version of Windows).

- If you have installed the 32-bit version of FluidLAB LibWaLi you will see the following window:



C:\Program Files\FluidLAB\LibWaLi

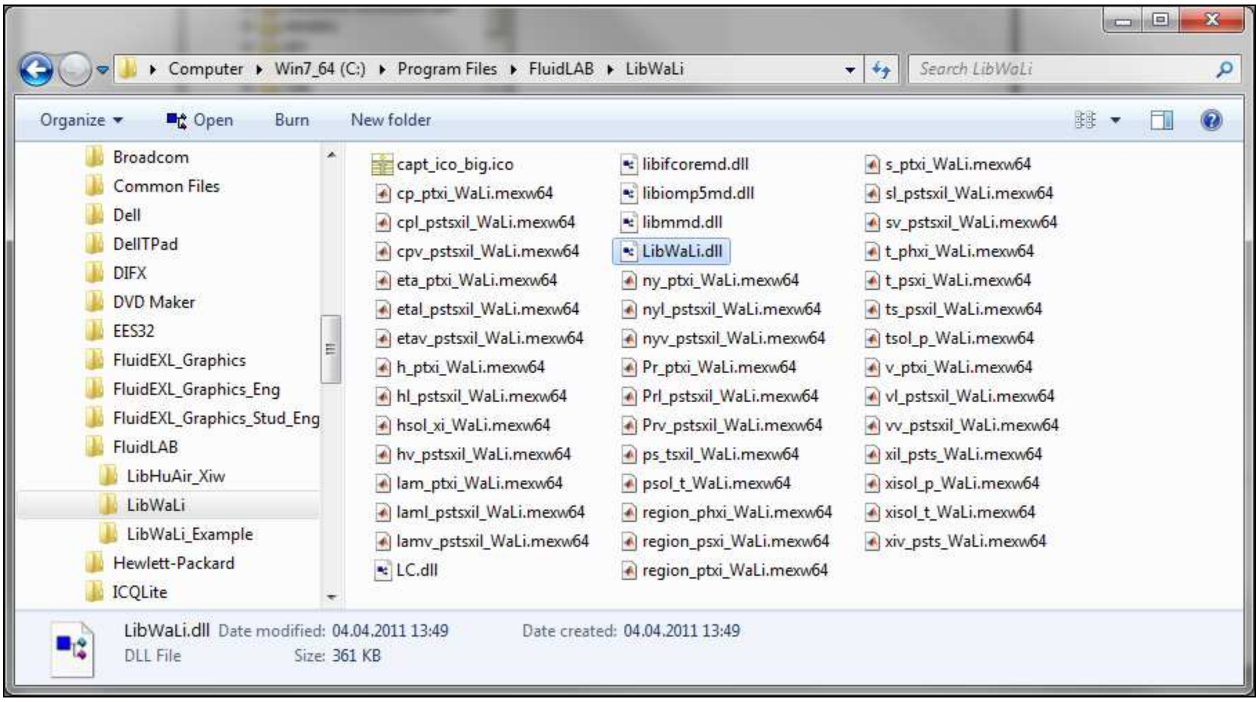
Administrator
All Users
Default User
Kretzschmar
lit05
Stoecker
Student
SWEEPUPD
Weidner
DRIVERS
EES
EES32
I386
installs
My Documents
Program Files
EES32
FluidLAB
LibWaLi
LibWaLi_Example

advapi32.dll
cp_ptxi_WaLi.mexw32
cpl_pstsxil_WaLi.mexw32
cpv_pstsxil_WaLi.mexw32
Dformd.dll
Dforrt.dll
eta_ptxi_WaLi.mexw32
etal_pstsxil_WaLi.mexw32
etav_pstsxil_WaLi.mexw32
h_ptxi_WaLi.mexw32
hl_pstsxil_WaLi.mexw32
hsol_xi_WaLi.mexw32
hv_pstsxil_WaLi.mexw32
INSTALL.LOG
lam_ptxi_WaLi.mexw32
laml_pstsxil_WaLi.mexw32
lamv_pstsxil_WaLi.mexw32
LC.dll
LibWaLi.dll
msvcp60.dll
msvcrt.dll
ny_ptxi_WaLi.mexw32
nyl_pstsxil_WaLi.mexw32
nyv_pstsxil_WaLi.mexw32
Pr_ptxi_WaLi.mexw32
Prl_pstsxil_WaLi.mexw32
Prv_pstsxil_WaLi.mexw32
ps_tsxil_WaLi.mexw32
psol_t_WaLi.mexw32
region_phxi_WaLi.mexw32
region_psxi_WaLi.mexw32
region_ptxi_WaLi.mexw32
s_ptxi_WaLi.mexw32
sl_pstsxil_WaLi.mexw32
sv_pstsxil_WaLi.mexw32
t_phxi_WaLi.mexw32
t_psxi_WaLi.mexw32
ts_psxil_WaLi.mexw32
tsol_p_WaLi.mexw32
UNWISE.EXE
UNWISE.INI
v_ptxi_WaLi.mexw32
vl_pstsxil_WaLi.mexw32
vv_pstsxil_WaLi.mexw32
xil_psts_WaLi.mexw32
xisol_p_WaLi.mexw32
xisol_t_WaLi.mexw32
xiv_psts_WaLi.mexw32

48 objects (Disk free space: 1.24 GB) 3.61 MB

Figure 2.6: Contents of the folder "LibWaLi" (32-bit version)

- If you have installed the 64-bit version of FluidLAB LibWaLi you will see the following window:


Computer ▸ Win7_64 (C:) ▸ Program Files ▸ FluidLAB ▸ LibWaLi
Search LibWaLi
Organize Open Burn New folder

Broadcom
Common Files
Dell
DellTPad
DIFX
DVD Maker
EES32
FluidEXL_Graphics
FluidEXL_Graphics_Eng
FluidEXL_Graphics_Stud_Eng
FluidLAB
LibHuAir_Xiw
LibWaLi
LibWaLi_Example
Hewlett-Packard
ICQLite

capt_ico_big.ico
cp_ptxi_WaLi.mexw64
cpl_pstsxil_WaLi.mexw64
cpv_pstsxil_WaLi.mexw64
eta_ptxi_WaLi.mexw64
etal_pstsxil_WaLi.mexw64
etav_pstsxil_WaLi.mexw64
h_ptxi_WaLi.mexw64
hl_pstsxil_WaLi.mexw64
hsol_xi_WaLi.mexw64
hv_pstsxil_WaLi.mexw64
lam_ptxi_WaLi.mexw64
laml_pstsxil_WaLi.mexw64
lamv_pstsxil_WaLi.mexw64
LC.dll
libifcoremd.dll
libiomp5md.dll
libmmd.dll
LibWaLi.dll
ny_ptxi_WaLi.mexw64
nyl_pstsxil_WaLi.mexw64
nyv_pstsxil_WaLi.mexw64
Pr_ptxi_WaLi.mexw64
Prl_pstsxil_WaLi.mexw64
Prv_pstsxil_WaLi.mexw64
ps_tsxil_WaLi.mexw64
psol_t_WaLi.mexw64
region_phxi_WaLi.mexw64
region_psxi_WaLi.mexw64
region_ptxi_WaLi.mexw64
s_ptxi_WaLi.mexw64
sl_pstsxil_WaLi.mexw64
sv_pstsxil_WaLi.mexw64
t_phxi_WaLi.mexw64
t_psxi_WaLi.mexw64
ts_psxil_WaLi.mexw64
tsol_p_WaLi.mexw64
v_ptxi_WaLi.mexw64
vl_pstsxil_WaLi.mexw64
vv_pstsxil_WaLi.mexw64
xil_psts_WaLi.mexw64
xisol_p_WaLi.mexw64
xisol_t_WaLi.mexw64
xiv_psts_WaLi.mexw64

LibWaLi.dll Date modified: 04.04.2011 13:49 Date created: 04.04.2011 13:49
DLL File Size: 361 KB

Figure 2.7: Contents of the folder "LibWaLi" (64-bit version)

If you have installed the 32-bit version of LibWaLi you will now have to copy the following files into the directory

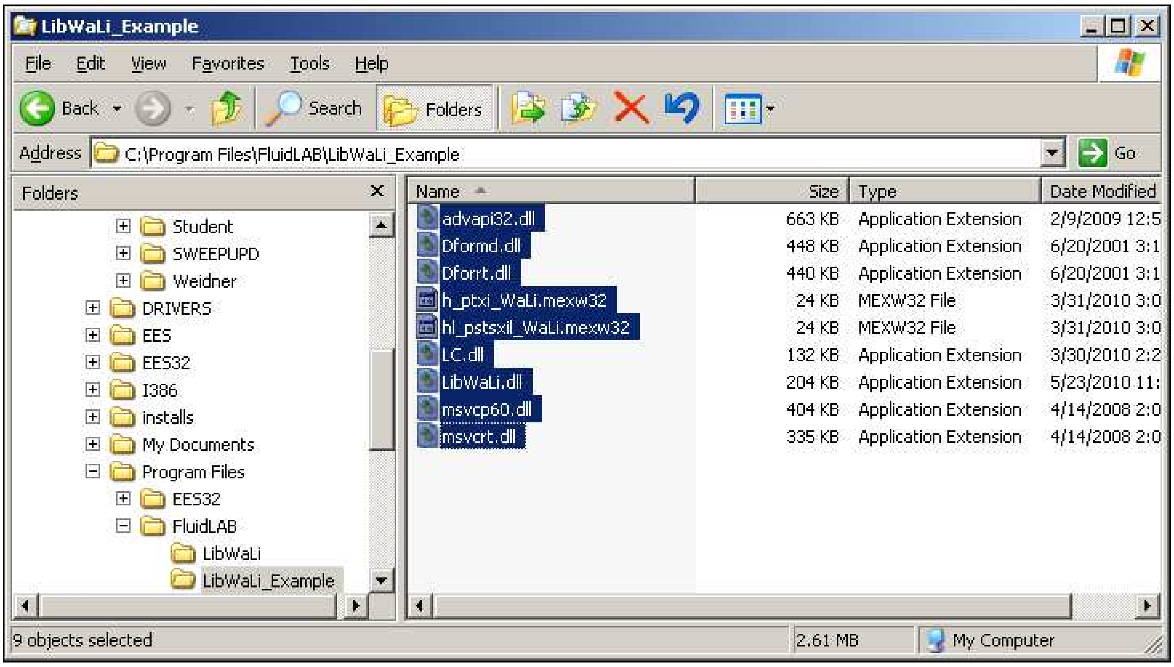
"C:\Program Files\FluidLAB\LibWaLi_Example" (for English version of Windows)
"C:\Programme\FluidLAB\LibWaLi_Example" (for German version of Windows)

in order to calculate the functions h = f(p,t,ξ) and h^l = f(p_s,t_s,ξ^1).

- The following files are needed:
  - "advapi32.dll"
  - "Dformd.dll"
  - "Dforrt.dll"
  - "h_ptxi_WaLi.mexw32"
  - "hl_pstsxil_WaLi.mexw32"
  - "LC.dll"
  - "LibWaLi.dll"
  - "msvcp60.dll"
  - "msvcrt.dll"
- Click the file "h_ptxi_WaLi.mexw32", then click "Edit" in the upper menu bar and select "Copy".
- Switch into the directory

  "C:\Program Files\FluidLAB\LibWaLi_Example" (for English version of Windows)
  "C:\Programme\FluidLAB\LibWaLi_Example" (for German version of Windows),

  click "Edit" and select "Paste".
- Repeat these steps in order to copy the other files listed above. You may also select all the above-named files and then copy them as a group (press the Control button to enable multiple markings).
- You will see the following window:



LibWaLi_Example

Address C:\Program Files\FluidLAB\LibWaLi_Example

Name	Size	Type	Date Modified
advapi32.dll	663 KB	Application Extension	2/9/2009 12:5
Dformd.dll	448 KB	Application Extension	6/20/2001 3:1
Dforrt.dll	440 KB	Application Extension	6/20/2001 3:1
h_ptxi_WaLi.mexw32	24 KB	MEXW32 File	3/31/2010 3:0
hl_pstsxil_WaLi.mexw32	24 KB	MEXW32 File	3/31/2010 3:0
LC.dll	132 KB	Application Extension	3/30/2010 2:2
LibWaLi.dll	204 KB	Application Extension	5/23/2010 11:
msvcp60.dll	404 KB	Application Extension	4/14/2008 2:0
msvcrt.dll	335 KB	Application Extension	4/14/2008 2:0

9 objects selected | 2.61 MB | My Computer

Figure 2.8: Contents of the folder "LibWaLi_Example" (32-bit version)

If you have installed the 64-bit version of LibWaLi you will now have to copy the following files into the directory

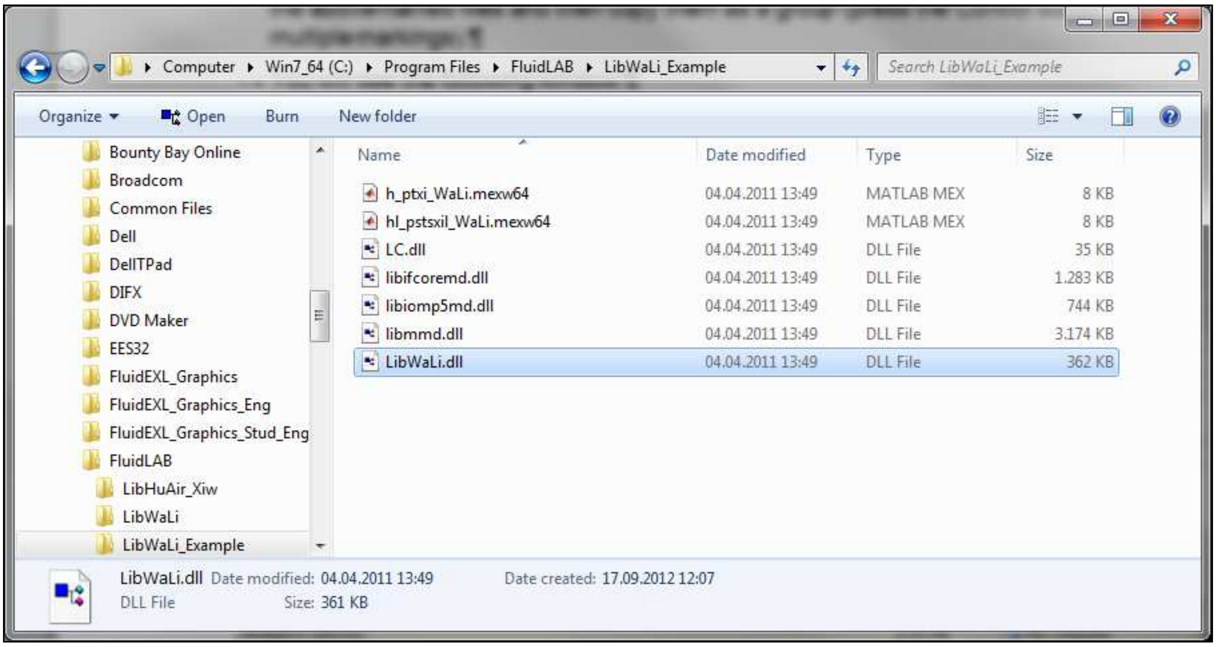
"C:\Program Files\FluidLAB\LibWaLi_Example" (for English version of Windows)
"C:\Programme\FluidLAB\LibWaLi_Example" (for German version of Windows)

in order to calculate the functions h = f(p,t,ξ) and h^l = f(p_s,t_s,ξ^1).

- The following files are needed:
  - "h_ptxi_WaLi.mexw64"
  - "hl_pstsxil_WaLi.mexw64"
  - "LC.dll"
  - "LibWaLi.dll"
  - "libifcoremd.dll"
  - "libiomp5.dll"
  - "libmmd.dll."
- Click the file "h_ptxi_WaLi.mexw64", then click "Edit" in the upper menu bar and select "Copy."
- Switch into the directory

  "C:\Program Files\FluidLAB\LibWaLi_Example" (for English version of Windows)
  "C:\Programme\FluidLAB\LibWaLi_Example" (for German version of Windows),

  click "Edit" and then "Paste."
- Repeat these steps in order to copy the other files listed above. You may also select all the above-named files and then copy them as a group (press the Control button to enable multiple markings).
- You will see the following window:



Name	Date modified	Type	Size
h_ptxi_WaLi.mexw64	04.04.2011 13:49	MATLAB MEX	8 KB
hl_pstsxil_WaLi.mexw64	04.04.2011 13:49	MATLAB MEX	8 KB
LC.dll	04.04.2011 13:49	DLL File	35 KB
libifcoremd.dll	04.04.2011 13:49	DLL File	1.283 KB
libiomp5md.dll	04.04.2011 13:49	DLL File	744 KB
libmmd.dll	04.04.2011 13:49	DLL File	3.174 KB
LibWaLi.dll	04.04.2011 13:49	DLL File	362 KB

LibWaLi.dll Date modified: 04.04.2011 13:49 Date created: 17.09.2012 12:07
DLL File Size: 361 KB

Figure 2.9: Contents of the folder "LibWaLi_Example"

- Start MATLAB® (if you have not started it already).
- Click the button marked in the next figure in order to open the folder "\LibWaLi_Example" in the "Current Directory" window.

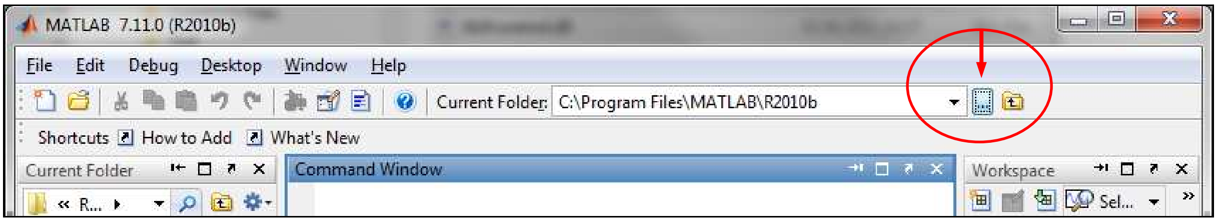
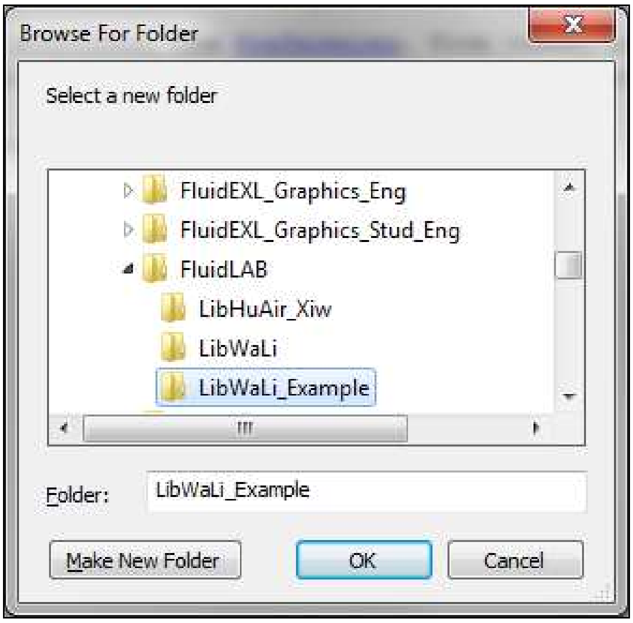

MATLAB 7.11.0 (R2010b)
File Edit Debug Desktop Window Help
Current Folder: C:\Program Files\MATLAB\R2010b
Shortcuts How to Add What's New
Current Folder
Command Window
Workspace
« R...
Sel...

Figure 2.10: Selection of the working directory

- Find and select the directory

  "C:\Program Files\FluidLAB\LibWaLi_Example" (for English version of Windows)
  "C:\Programme\FluidLAB\LibWaLi_Example" (for German version of Windows)

  in the menu which appears (see the following figure).


Browse For Folder

Select a new folder

Folder: LibWaLi_Example

Figure 2.11: Choosing the "LibWaLi_Example" folder

- Confirm your selection by clicking the "OK" button.
- First of all you need to create an M–File in MATLAB®. Within MATLAB® click "File", then select "New" and afterwards click "M-File" in MATLAB 2006 or earlier versions or click "Script" in MATLAB 2010.
- If the "Editor" window appears as a separate window, you can embed it into MATLAB® by clicking the insertion arrow (see next figure) in order to obtain a better view.

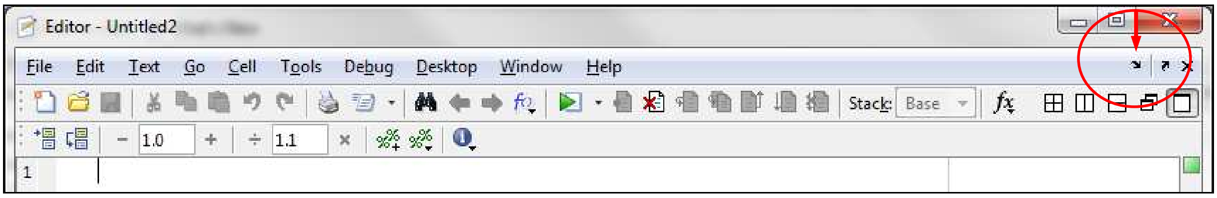
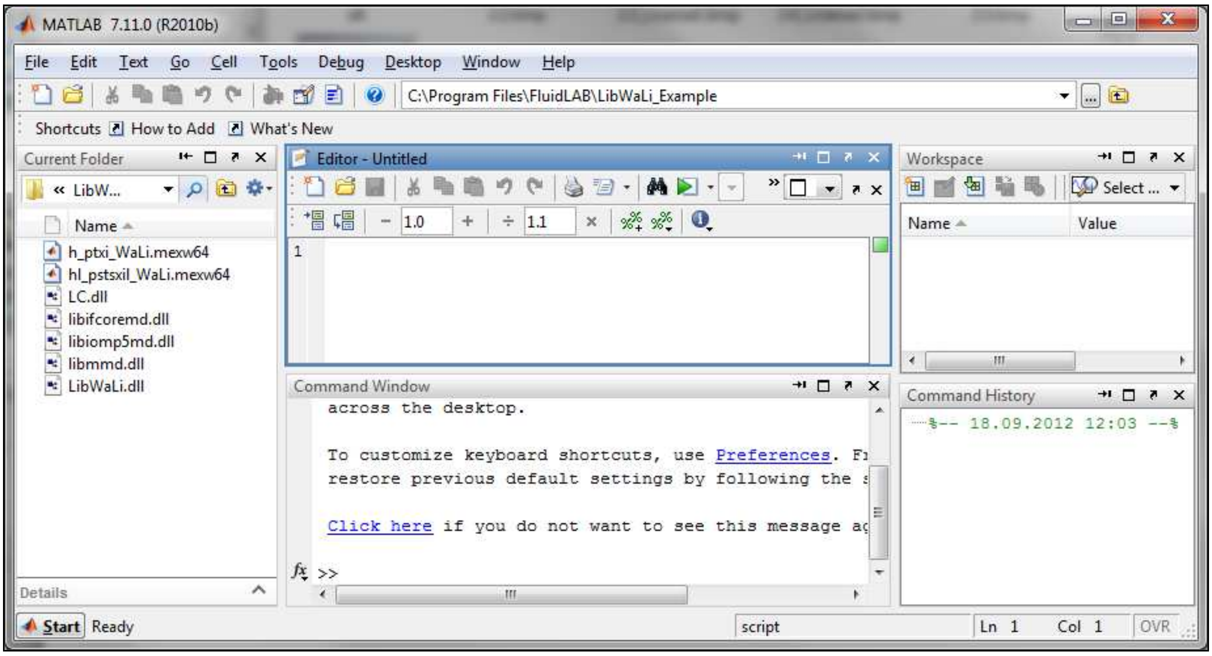


Figure 2.12: Embedding the "Editor" window

- In the figure below you will see the "Editor - Untitled" window.



```
across the desktop.

To customize keyboard shortcuts, use Preferences. Fr
restore previous default settings by following the s

Click here if you do not want to see this message ag

fx >>
```

```
%-- 18.09.2012 12:03 --%
```

Figure 2.13: Embedded "Editor" window

- Now type the following lines in the "Editor - Untitled" window:

Text to be written:	Explanation:
`% h_ptxi_WaLi.m`	file name as comment
`%%`	paragraph separation
`p=1; % pressure in bar`	declaration of the variables pressure, temperature and mass fraction H_20
`t=25; % temperature in °C`	
`xi=0.6; % mass fraction in kgH2O/kg`	
`%%`	paragraph separation
`h=h_ptxi_WaLi(p,t,xi)`	function call
`%%`	paragraph separation

- Remarks:

  - The program interprets the first line which starts with " % " to be a data description in "Current Directory"
  - Paragraph separations which are mandatory are being realised through " %%". By this, declaration of variables and calculation instructions are also being separated.
  - The words which are printed in green, start with "%" and stand behind the variables are comments. In fact they are not necessary but they are reasonable for your overview and comprehensibility.
  - You have to leave out the semicolons behind the numerical values if you wish to see the result for *h* and the input parameters as well.

The values of the function parameters in their corresponding units stand for:

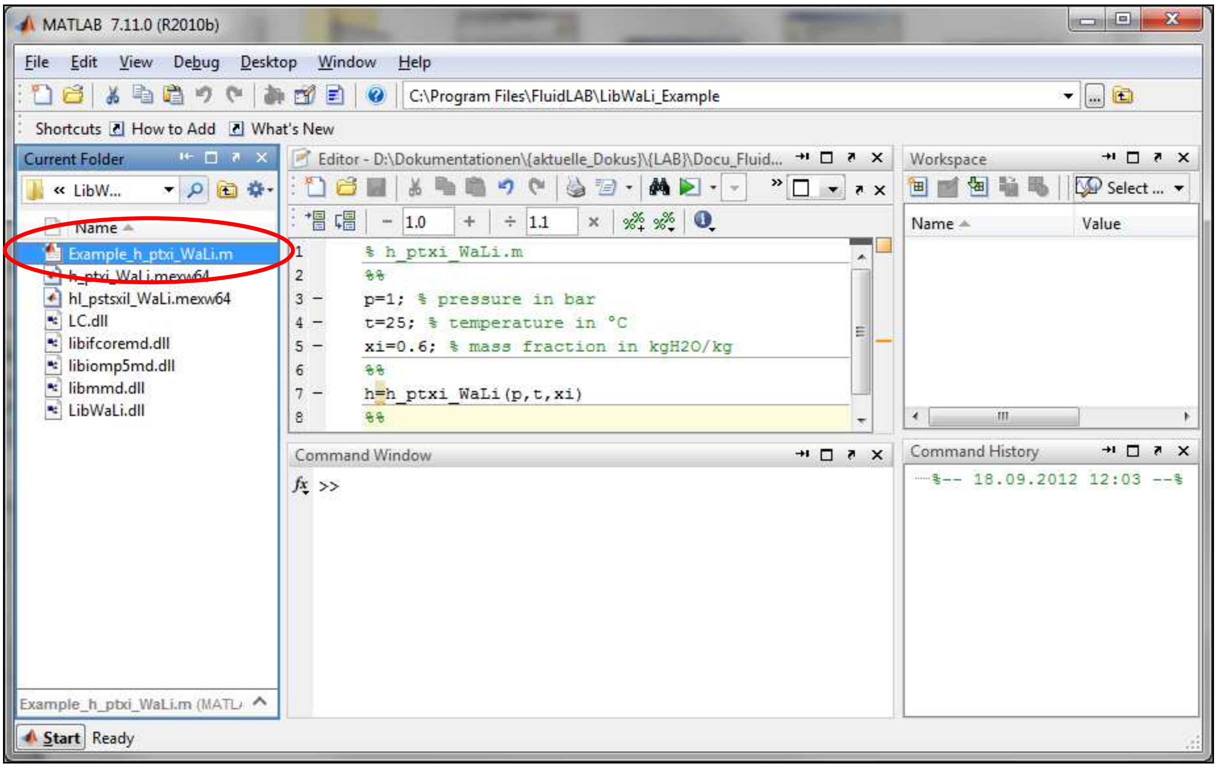
- First operand: Value for *p* = 1 bar
  (Range of validity: *p* = 0.00074 bar ... 10 bar)
- Second operand: Value for *t* = 25 °C
  (Range of validity: *t* = 0 °C ... 210 °C)

  - Third operand: Value for ξ = 0.6
    (Range of validity: ξ = 0.3 kg H_2O/kg …1.0 kg H_2O/kg)

- Save the "M-File" by clicking the "File" button and then click "Save As...".
- The menu "Save file as:" appears; In this menu, the folder name "LibWaLi_Example" must be displayed in the "Save in:" field
- Next to "File name" you have to type in "Example_h_ptxi_WaLi.m" and afterwards click the "Save" button.

Note.
The name of the example file has to be different in comparison to the name of the used function. For example, the file could not be named "h_ptxi_WaLi.m" in this case. Otherwise an error message will appear during the calculation.

- You will now see the following window:



```
% h_ptxi_WaLi.m
%%
p=1; % pressure in bar
t=25; % temperature in °C
xi=0.6; % mass fraction in kgH2O/kg
%%
h=h_ptxi_WaLi(p,t,xi)
%%
```

```
>>
```

```
%-- 18.09.2012 12:03 --%
```

Figure 2.14: "Example_h_ptxi_WaLi.m" M-file

- Within the "Current Directory" window the file "Example_h_ptxi_WaLi.m" appears.
- Right-click on this file and select "Run" in the menu which appears (see next image).

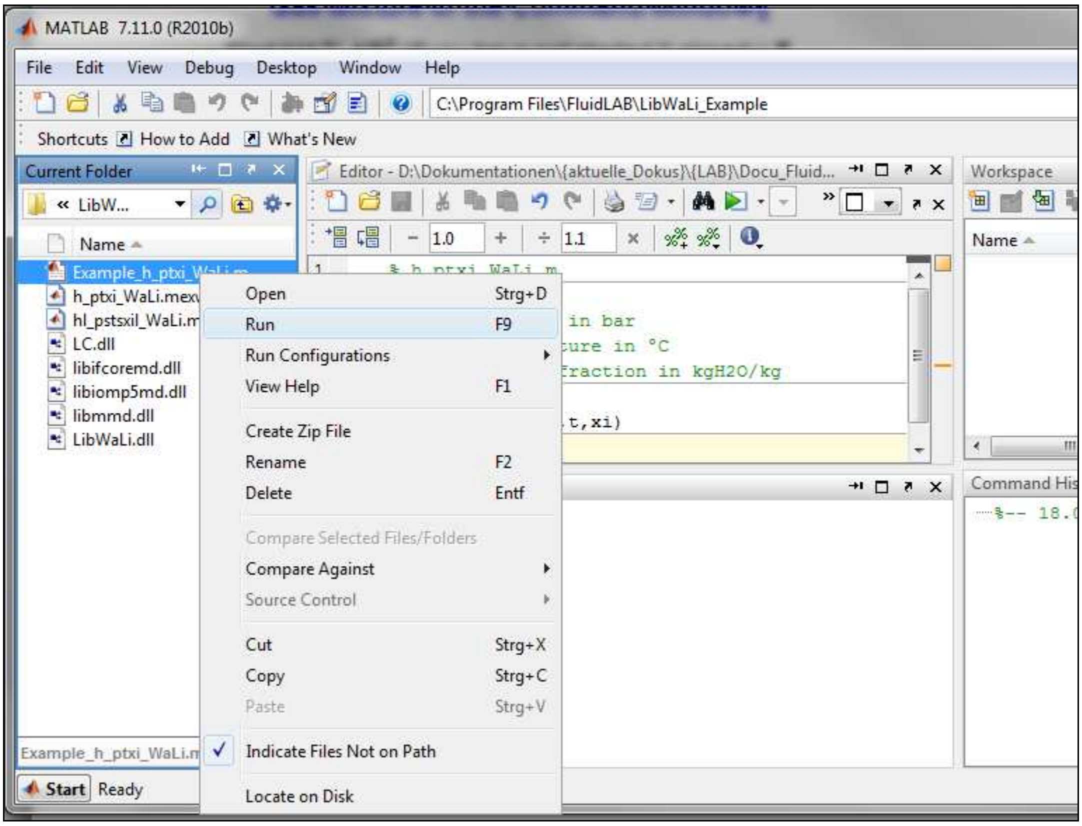
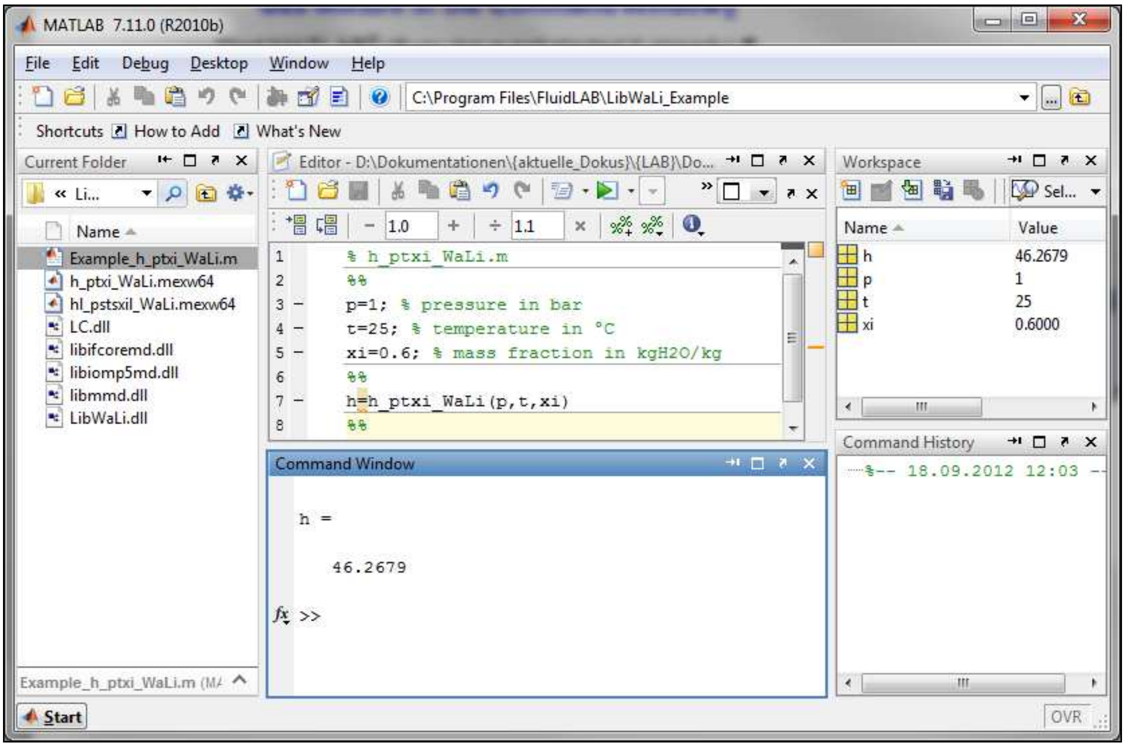


Figure 2.15: Running the "Example_h_ptxi_WaLi.m" M-file

- You will see the following window:



```
% h_ptxi_WaLi.m
%%
p=1; % pressure in bar
t=25; % temperature in °C
xi=0.6; % mass fraction in kgH2O/kg
%%
h=h_ptxi_WaLi(p,t,xi)
%%
```

```
h =

    46.2679

>>
```

Name	Value
h	46.2679
p	1
t	25
xi	0.6000

Figure 2.16: MATLAB$^{®}$ with calculated result

The result for *h* appears in the "Command Window".

⇒ The result in our sample calculation here is: "h = 46.2679". The corresponding unit is kJ/kg (see table of the property functions in Chapter 1).

To be able to calculate other values, you have to copy the associated mexw32 or mexw64 files as well because MATLAB® can only access functions that are located in the "Current Directory" window. The example calculated can be found in the directory

C:\Program Files\FluidLAB\LibWaLi_Example" (for English version of Windows)
C:\Programme\FluidLAB\LibWaLi_Example" (for German version of Windows),

and you may use it as a basis for further calculations using FluidLAB.

Hint!

If the input values are located outside the range of validity of LibWaLi, the calculation of the chosen function to be calculated function results in -1000. You can find more exact details on every function and its corresponding range of validity in the enclosed program documentation in Chapter 3.

2.4 Example: Calculation of the Specific Enthalpy *h* = f(*p*,*t*,ξ) for Water/Lithium Bromide in Command Window

- Please follow the instructions from page 2/7 to 2/10.
- Start MATLAB® (if you have not started it already).
- Click the button marked in the following figure in order to open the folder "\LibWaLi_Example" in the window "Current Directory".

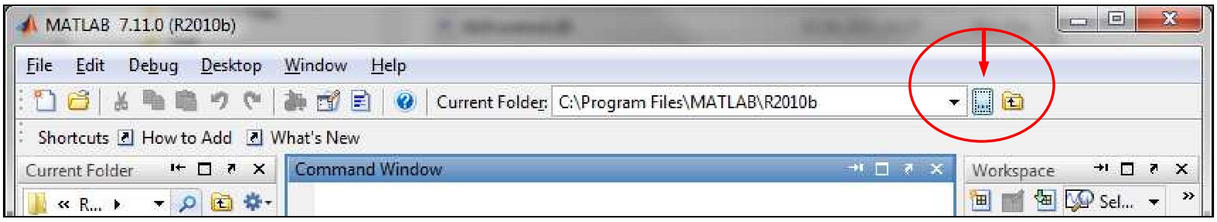

MATLAB 7.11.0 (R2010b)
File Edit Debug Desktop Window Help
Current Folder: C:\Program Files\MATLAB\R2010b
Shortcuts How to Add What's New
Current Folder
Command Window
Workspace
« R...
Sel...

Figure 2.17: Selection of the working directory

- Find and select the directory

  "C:\Program Files\FluidLAB\LibWaLi_Example" (for English version of Windows)
  "C:\Programme\FluidLAB\LibWaLi_Example" (for German version of Windows)

  in the menu that appears (see figure below).

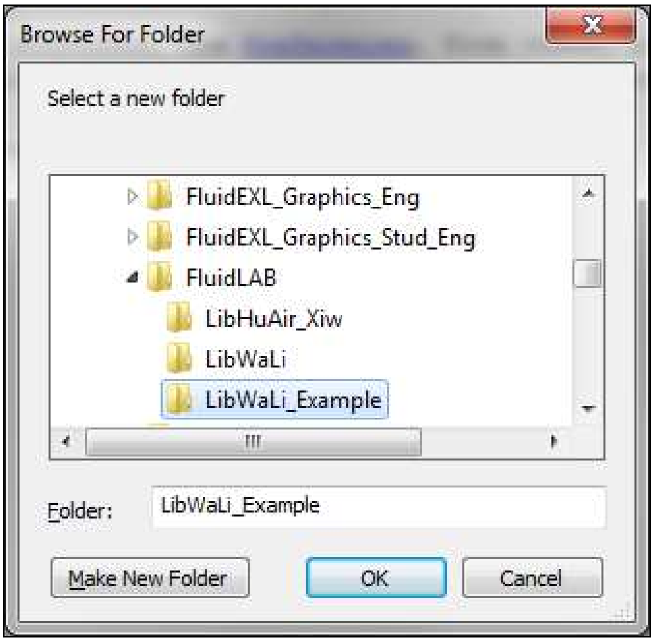
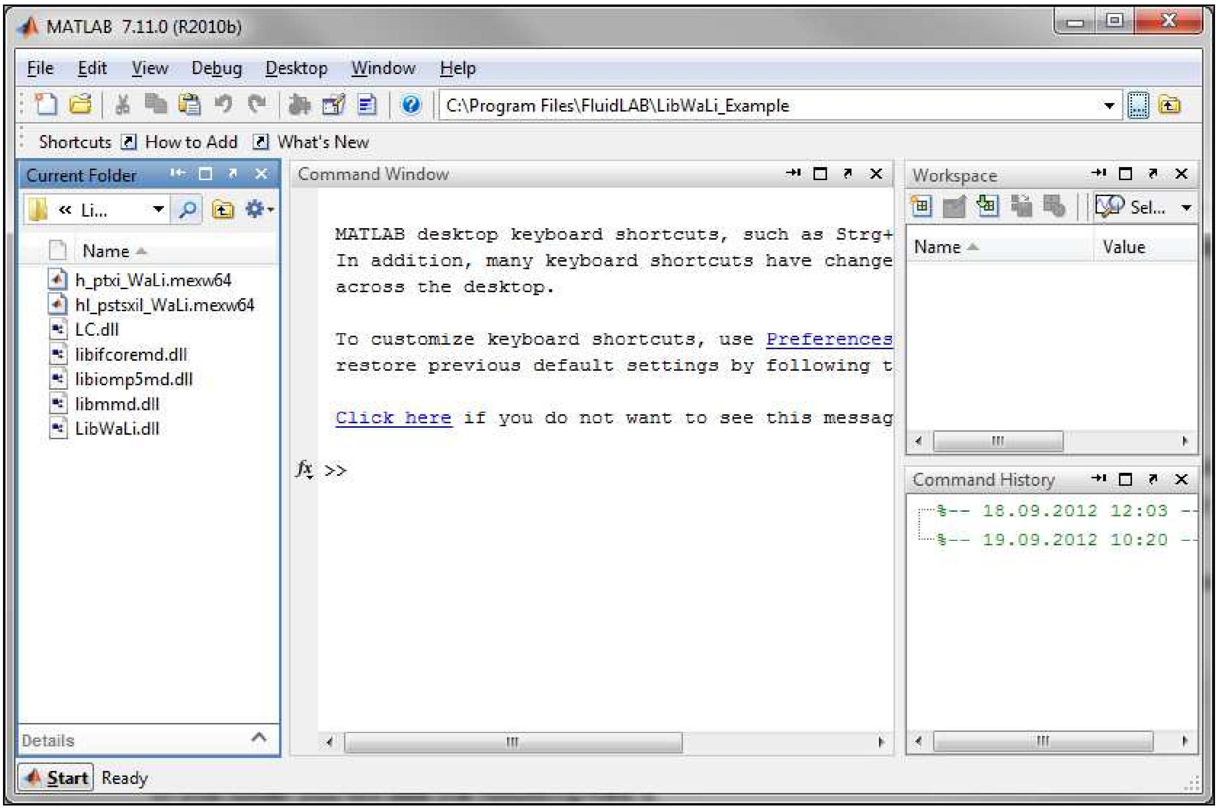


Figure 2.18: Choosing the "LibWaLi_Example" folder

- Confirm your selection by clicking the "OK" button.

- You will see the following window:



C:\Program Files\FluidLAB\LibWaLi_Example

h_ptxi_WaLi.mexw64
hl_pstsxil_WaLi.mexw64
LC.dll
libifcoremd.dll
libiomp5md.dll
libmmd.dll
LibWaLi.dll

```
MATLAB desktop keyboard shortcuts, such as Strg+
In addition, many keyboard shortcuts have change
across the desktop.

To customize keyboard shortcuts, use Preferences
restore previous default settings by following t

Click here if you do not want to see this messag

fx >>
```

```
%-- 18.09.2012 12:03 --
%-- 19.09.2012 10:20 --
```

Figure 2.19: MATLAB® with necessary files

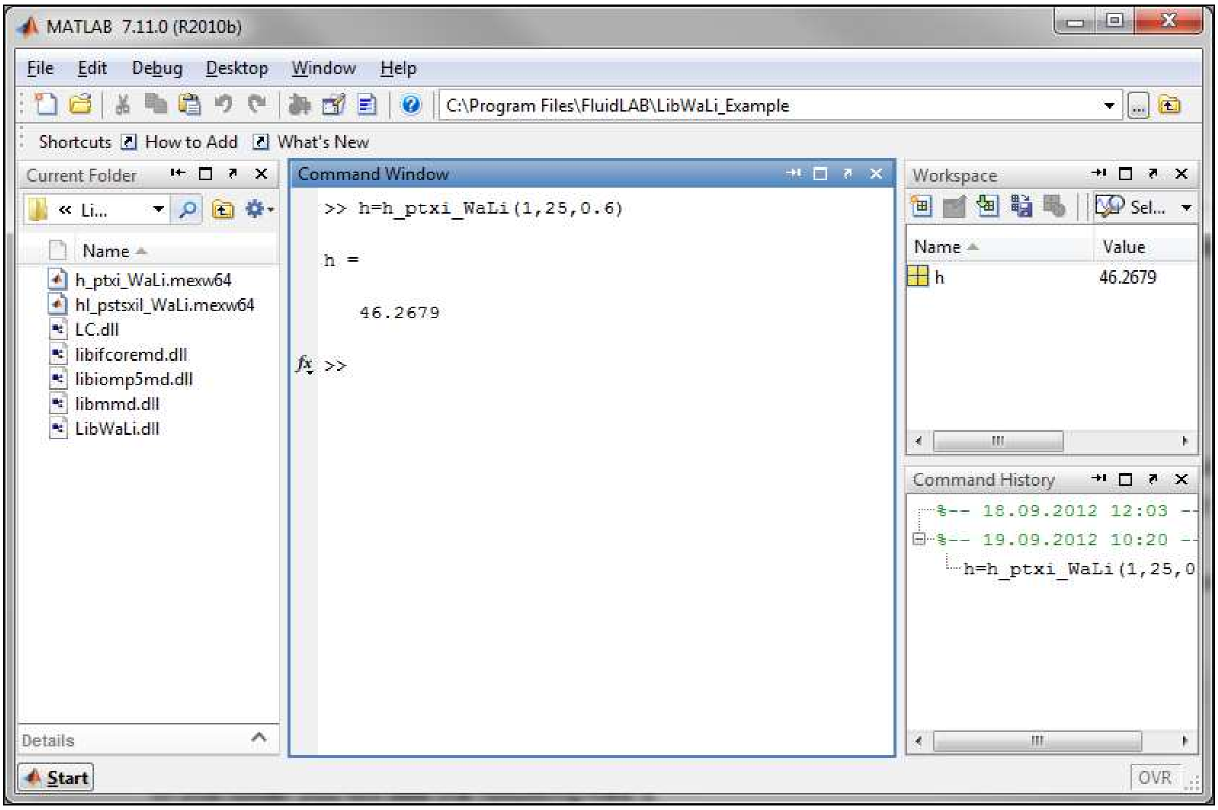
Corresponding to the table of property functions in Chapter 1 you have to call up the function "**h_ptxi_WaLi**" as follows for calculating h=f(p,t,ξ):

- Write "**h=h_ptxi_WaLi(1,25,0.6)**" within the "Command Window".

The values of the function parameters in their corresponding units stand for:

- First operand: Value for p = 1 bar
  (Range of validity: p = 0.00074 bar ... 10 bar)
- Second operand: Value for t = 25 °C
  (Range of validity: t = 0°C ... 210 °C)
- Third operand: Value for ξ = 0.6
  (Range of validity: ξ = 0.3 kg H_2O/kg …1.0 kg H_2O/kg)

- Confirm your entry by pressing the "ENTER" button.
- You will see the following window:



C:\Program Files\FluidLAB\LibWaLi_Example

Current Folder: « Li...

Name
- h_ptxi_WaLi.mexw64
- hl_pstsxil_WaLi.mexw64
- LC.dll
- libifcoremd.dll
- libiomp5md.dll
- libmmd.dll
- LibWaLi.dll

```
>> h=h_ptxi_WaLi(1,25,0.6)

h =

   46.2679

fx >>
```

Name	Value
h	46.2679

```
%-- 18.09.2012 12:03 --
%-- 19.09.2012 10:20 --
  h=h_ptxi_WaLi(1,25,0
```

Figure 2.20: MATLAB® with calculated result

In the "Command Window" you will see the result "h = 46.2679". The corresponding unit is kJ/kg (see table of the property functions in chapter 1).

To be able to calculate other values, you will have to copy the respective mexw32 or mexw64 files into the working directory as well, because MATLAB® can only access functions that are located in the "Current Directory" window. The example calculated can be found in the directory

C:\Program Files\FluidLAB\LibWaLi_Example"	(for English version of Windows)
C:\Programme\FluidLAB\LibWaLi_Example"	(for German version of Windows),

and you may use it as a basis for further calculations using FluidLAB.

Note:

If the input values are located outside the range of validity, the result for the chosen function to be calculated results in -1000. You can find more exact details on every function and its corresponding range of validity in the enclosed program documentation in Chapter 3.

2.5 Example: Calculation of the Specific Enthalpy $h^l = f(p_s, t_s, \xi^l)$ of Saturated Liquid for Water/Lithium Bromide in an M-File

- Please follow the instructions from page 2/7 to 2/10.
- Start MATLAB® (if you have not started it already).
- Click the button marked in the following figure in order to open the folder "\LibWaLi_Example" in the window "Current Directory".

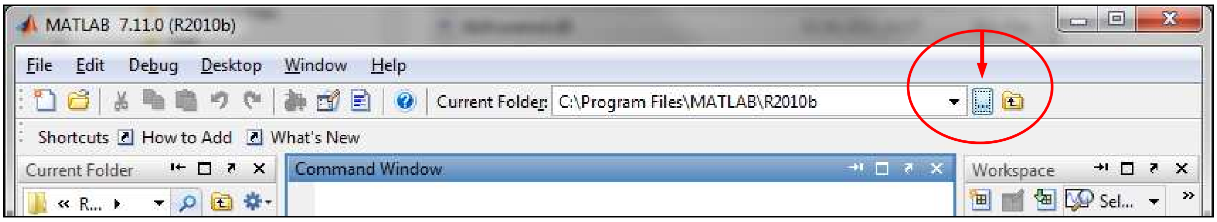
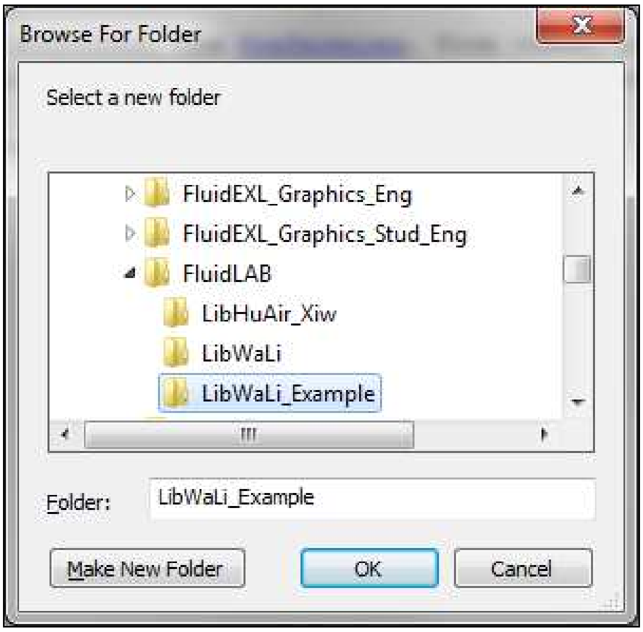

MATLAB 7.11.0 (R2010b)
File Edit Debug Desktop Window Help
Current Folder: C:\Program Files\MATLAB\R2010b
Shortcuts How to Add What's New
Current Folder
Command Window
Workspace
« R...
Sel...

Figure 2.21: Selection of the working directory

- Find and select the directory

  "C:\Program Files\FluidLAB\LibWaLi_Example" (for English version of Windows)
  "C:\Programme\FluidLAB\LibWaLi_Example" (for German version of Windows)

  in the menu which appears (see the following figure).



Folder: LibWaLi_Example

Figure 2.22: Choosing the "LibWaLi_Example" folder

- Confirm your selection by clicking the "OK" button.

- First of all you need to create an M–File in MATLAB®. Within MATLAB® click "File", then select "New" and afterwards click "M-File" in MATLAB 2006 or earlier versions or click "Script" in MATLAB 2010.
- If the "Editor" window appears as a separate window, you can embed it into MATLAB® by clicking the insertion arrow (see next figure) in order to obtain a better view.

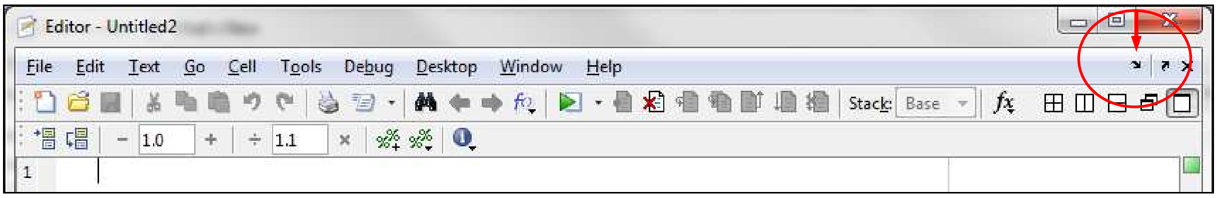
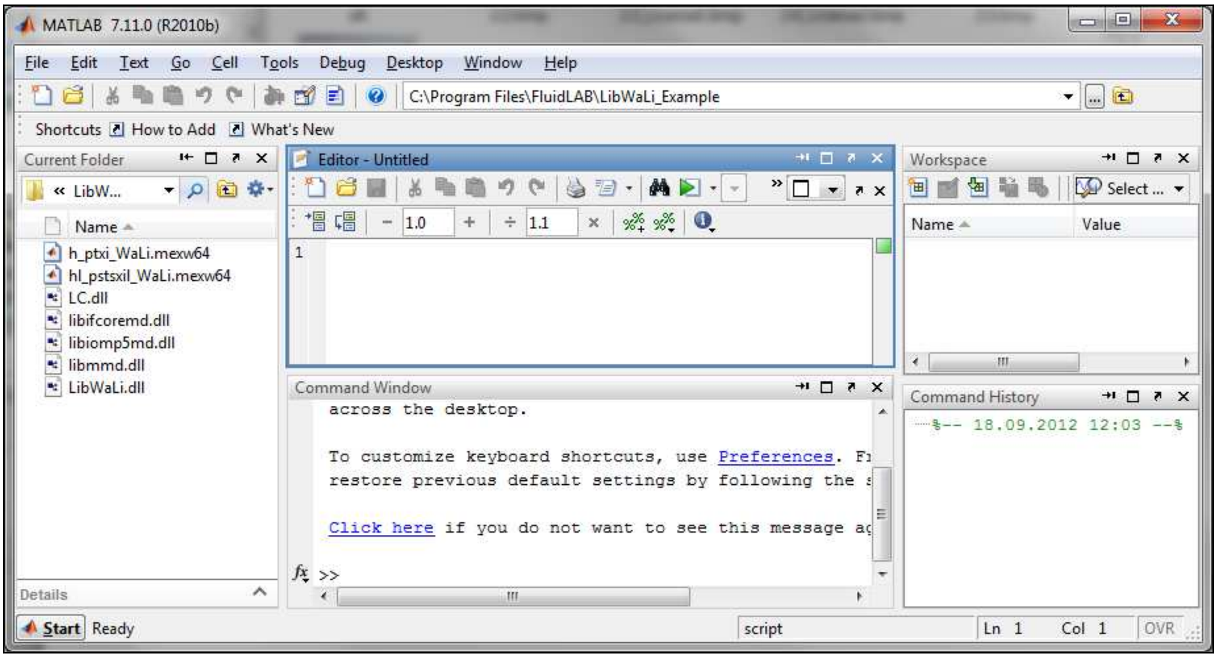


Figure 2.23: Embedding the "Editor" window

- In the figure below you will see the "Editor - Untitled" window.



```
across the desktop.

To customize keyboard shortcuts, use Preferences. Fi
restore previous default settings by following the :

Click here if you do not want to see this message ac

fx >>
```

```
%-- 18.09.2012 12:03 --%
```

Figure 2.24: Embedded "Editor" window

- Now type the following lines in the "Editor - Untitled" window:

Text to be written:	Explanation:
`% hl_pstsxil_WaLi.m`	file name as comment
`%%`	paragraph separation
`ps=1; % vapor pressure in bar`	declaration of the variables vapor pressure, saturation temperature and mass fraction H_20 of saturated liquid
`ts=-1; % saturation temperature in °C`	
`xil=0.6; % mass fraction saturated liquid in kgH2O/kg`	
`%%`	paragraph separation
`hl=hl_pstsxil_WaLi(ps,ts,xil)`	function call
`%%`	paragraph separation

- Remarks:

  - The program interprets the first line, starting with "%," to be a data description in "Current Directory."
  - Paragraph separations which are mandatory are marked with "%%". This also serves to separate the declaration of variables and calculation instructions.
  - The words which are printed in green, start with "%" and come after the variables are comments. They are not in fact absolutely necessary, but they are very helpful for your overview and to make the process more easily understood.
  - Omit the semicolons after the numerical values if you wish to see the result for *hl* and the input parameters.

The values of the function parameters in their corresponding units stand for:

- First operand: Value for p_s = 1 bar

  (Range of validity: p_S = 0.00074 bar ... 10 bar)

- Second operand: Value for t_s = -1

  When calculating the specific enthalpy of the saturated liquid it suffices to define two parameters (either t_S and ξ^l or p_S and ξ^l or p_S and t_S). It is necessary to enter -1 as a pro-forma value for the value which is not given. In case p_S, t_S, and ξ^l are entered, the program tests whether p_S, t_S, and ξ^l fulfil the vapor pressure curve. If this is not the case the calculation of the chosen function to be calculated results in -1000.

  Possible input variants: $h^l = f(-1, t_S, \xi^l)$
  $h^l = f(p_S, -1, \xi^l)$
  $h^l = f(p_S, t_S, -1)$
  $h^l = f(p_S, t_S, \xi^l)$

  (Range of validity: t_S = 0 °C ... 210 °C)

- Third operand: Value for ξ^l = 0.6 kg H_2O/kg

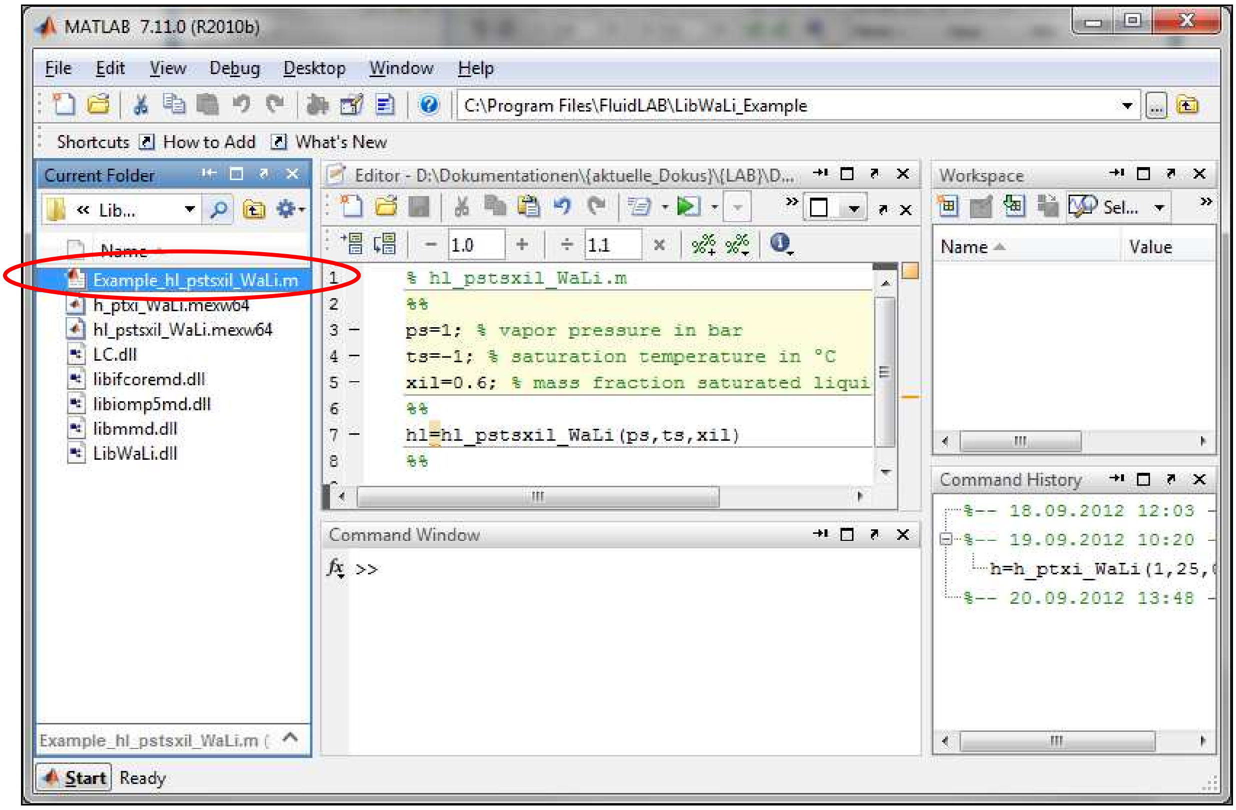
  (Range of validity: ξ^l = 0.3 kg H_2O/kg ... 1.0 kg H_2O/kg)

- Save the "M-File" by clicking the "File" button and then click "Save As...".
- The menu "Save file as:" appears; In this menu, the folder name "LibWaLi_Example" must be displayed in the "Save in:" field.
- Next to "File name" you have to type in "Example_hl_pstsxil_WaLi.m" and afterwards click the "Save" button.

Note.

The name of the example file has to be different in comparison to the name of the used function. For example, the file could not be named "hl_pstsxil_WaLi.m" in this case. Otherwise an error message will appear during the calculation.

- You will see the following window:


MATLAB 7.11.0 (R2010b)
File Edit View Debug Desktop Window Help
C:\Program Files\FluidLAB\LibWaLi_Example
Shortcuts How to Add What's New
Current Folder
Lib...
Name
Example_hl_pstsxil_WaLi.m
h_ptxi_WaLi.mexw64
hl_pstsxil_WaLi.mexw64
LC.dll
libifcoremd.dll
libiomp5md.dll
libmmd.dll
LibWaLi.dll
Example_hl_pstsxil_WaLi.m (
Editor - D:\Dokumentationen\{aktuelle_Dokus}\{LAB}\D...
1.0
1.1
Command Window
Workspace
Sel...
Name
Value
Command History
Start
Ready

```
% hl_pstsxil_WaLi.m
%%
ps=1; % vapor pressure in bar
ts=-1; % saturation temperature in °C
xil=0.6; % mass fraction saturated liqui
%%
hl=hl_pstsxil_WaLi(ps,ts,xil)
%%
```

```
>>
```

```
%-- 18.09.2012 12:03 -
%-- 19.09.2012 10:20 -
   h=h_ptxi_WaLi(1,25,(
%-- 20.09.2012 13:48 -
```

Figure 2.25: "Example_hl_pstsxil_WaLi.m" M-file

- Within the "Current Directory" window, the file "Example_hl_pstsxil_WaLi.m" appears.
- Right-click on this file and select "Run" in the menu which appears (see next image).

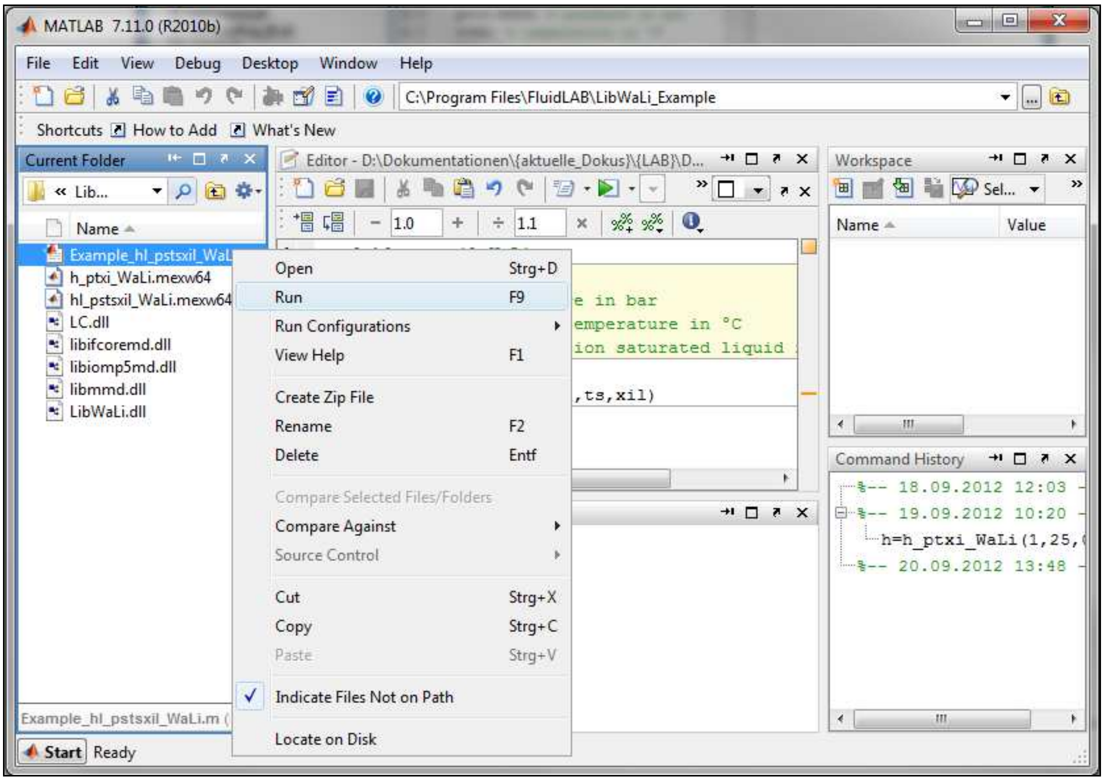
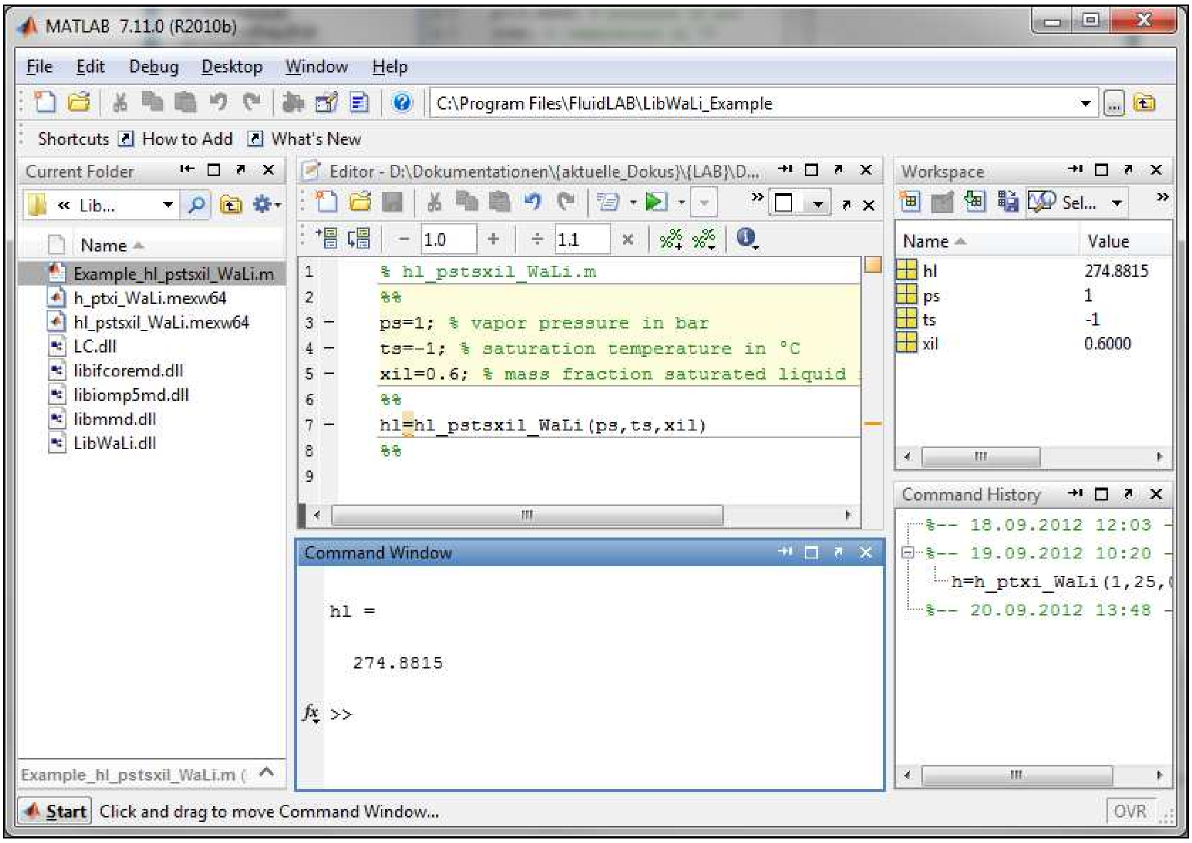


Figure 2.26: Running the "Example_hl_pstsxil_WaLi.m" M-file

- You will see the following window:


MATLAB 7.11.0 (R2010b)
File Edit Debug Desktop Window Help
C:\Program Files\FluidLAB\LibWaLi_Example
Shortcuts How to Add What's New
Current Folder
« Lib...
Name
Example_hl_pstsxil_WaLi.m
h_ptxi_WaLi.mexw64
hl_pstsxil_WaLi.mexw64
LC.dll
libifcoremd.dll
libiomp5md.dll
libmmd.dll
LibWaLi.dll
Example_hl_pstsxil_WaLi.m (
Editor - D:\Dokumentationen\{aktuelle_Dokus}\{LAB}\D...
1.0
1.1
Command Window
Workspace
Sel...
Command History
Start
Click and drag to move Command Window...
OVR

```
% hl_pstsxil_WaLi.m
%%
ps=1; % vapor pressure in bar
ts=-1; % saturation temperature in °C
xil=0.6; % mass fraction saturated liquid
%%
hl=hl_pstsxil_WaLi(ps,ts,xil)
%%
```

```
hl =

  274.8815

>>
```

Name	Value
hl	274.8815
ps	1
ts	-1
xil	0.6000

```
%-- 18.09.2012 12:03 -
%-- 19.09.2012 10:20 -
   h=h_ptxi_WaLi(1,25,(
%-- 20.09.2012 13:48 -
```

Figure 2.27: MATLAB® with calculated result

In the "Command Window" you will see the result "hl = 274.8815". The corresponding unit is kJ/kg (see table of the property functions in Chapter 1).

To be able to calculate other values, you have to copy the associated mexw32 or mexw64 files as well because MATLAB® can only access functions that are located in the "Current Directory" window. The example calculated can be found in the directory

C:\Program Files\FluidLAB\LibWaLi_Example" (for English version of Windows)
C:\Programme\FluidLAB\LibWaLi_Example" (for German version of Windows),

and you may use it as a basis for further calculations using FluidLAB.

Note:

If the input values are located outside the range of validity of LibWaLi, the calculation of the chosen function to be calculated results in -1000. You can find more exact details on every function and its corresponding range of validity in the enclosed program documentation in Chapter 3.

2.6 Removing FluidLAB LibWaLi

To remove the property library LibWaLi from your hard drive in Windows®, click "Start" in the Windows® task bar, select "Settings" and click "Control Panel".

Now double-click on "Add or Remove Programs". In the list box of the "Add or Remove Programs" window that appears select "FluidLAB LibHuAirProp" by clicking on it and click the "Change/Remove" button.

In the following dialog box click "Automatic" and then click the "Next >" button.

Confirm the following menu "Perform Uninstall" by clicking the "Finish" button.

Finally, close the "Add or Remove Programs" and "Control Panel" windows.

Now, FluidLAB has been removed.

If there is no library other than LibWaLi installed, the directory "FluidLAB" will be removed as well.

3. Program Documentation

Specific Isobaric Heat Capacity $c_p = f(p,t,\xi)$

Function Name:	**cp_ptxi_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION CP_PTXI_WALI(P,T,XI)** REAL*8 P,T,XI
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_CP_PTXI_WALI(CP,P,T,XI)** REAL*8 CP,P,T,XI

Input Values:

P - Pressure p in bar

T - Temperature t in °C

XI - Mass fraction of water ξ in (kg H_2O)/(kg mixture)

Result

CP_PTXI_WALI, **CP** or **cp_ptxi_WaLi** - Specific isobaric heat capacity c_p in kJ/(kg K)

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints for wet steam

In the wet steam region the calculation is carried out for saturated liquid and saturated steam only. The value -1000 will be shown for the wet steam region lying between.

Results for wrong input values

Result **CP_PTXI_WALI = -1000**, **CP = -1000** or **cp_ptxi_WaLi = -1000** for input values:

p > 10 bar or p < 0.00074 bar or
t > 210 °C or t < 0 °C or
$\xi_s > \xi_{sol}(t)$ or
ξ > 1.0 kg/kg or ξ < 0.3 kg/kg
Points in the wet steam region between saturated liquid and saturated steam

References: [2], [21], [23]

Specific Isobaric Heat Capacity of Saturated Liquid $c_p^l = f(p_s, t_s, \xi^l)$

Function Name:	**cpl_pstsxil_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION CPL_WALI(PS,TS,XIL)** REAL*8 PS,TS,XIL
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_CPL_WALI(CPL,PS,TS,XIL)** REAL*8 CPL,PS,TS,XIL

Input Values:

PS - Vapor pressure p_s in bar

TS - Saturation temperature t_s in °C

XIL - Mass fraction of water as saturated liquid ξ^l in (kg H_2O)/(kg mixture)

Result

CPL_WALI, **CPL** or **cpl_pstsxil_WaLi** - Specific isobaric heat capacity of the saturated liquid c_p^l in kJ / (kg K)

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints of the input variants

For calculating the specific isobaric heat capacity of the saturated liquid it is necessary to define two parameters (either t_S and ξ' or p_S and ξ' or p_S and t_S) and –1 must be entered as a pro-forma value for the value which is not given. When p_S, t_S, and ξ' are entered, the program considers p_S, t_S, and ξ' to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$c_p^l = f(-1, t_s, \xi^l)$

$c_p^l = f(p_s, -1, \xi^l)$

$c_p^l = f(p_s, t_s, -1)$

$c_p^l = f(p_s, t_s, \xi^l)$

Results for wrong input values

Result **CPL_WALI = −1000**, **CPL = −1000** or **cpl_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

References: [2], [21], [23]

Specific Isobaric Heat Capacity of Saturated Steam $c_p^v = f(p_s, t_s, \xi^l)$

Function Name:	**cpv_pstsxil_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION CPV_WALI(PS,TS,XIL)** REAL*8 PS,TS,XIL
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_CPV_WALI(CPV,PS,TS,XIL)** REAL*8 CPV,PS,TS,XIL

Input Values:

PS - Vapor pressure p_s in bar

TS - Saturation temperature t_s in °C

XIL - Mass fraction of water as saturated liquid ξ^l in (kg H_2O)/(kg mixture)

Result

CPV_WALI, **CPV** or **cpv_pstsxil_WaLi** - Specific isobaric heat capacity of saturated steam c_p^v in kJ/(kg K)

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints of the input variants

For calculating the specific isobaric heat capacity of saturated steam it is necessary to define two parameters (either t_s and ξ' or p_s and ξ' or p_s and t_s) and –1 must be entered as a pro-forma value for the value which is not given. When p_s, t_s, and ξ' are entered, the program considers p_s, t_s, and ξ' to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$c_p^v = f(-1, t_s, \xi^l)$

$c_p^v = f(p_s, -1, \xi^l)$

$c_p^v = f(p_s, t_s, -1)$

$c_p^v = f(p_s, t_s, \xi^l)$

Results for wrong input values

Result **CPV_WALI = −1000**, **CPV = −1000** or **cpv_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

References: [2], [23]

Dynamic Viscosity $\eta = f(p,t,\xi)$

Function Name:	**eta_ptxi_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION ETA_PTXI_WALI(P,T,XI)** REAL*8 P,T,XI
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_ETA_PTXI_WALI(ETA,P,T,XI)** REAL*8 ETA,P,T,XI

Input Values:

P - Pressure p in bar

T - Temperature t in °C

XI - Mass fraction of water ξ in (kg H_2O)/(kg mixture)

Result

ETA_PTXI_WALI, **ETA** or **eta_ptxi_WaLi** – Dynamic viscosity η in Pa s

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints for wet steam

In the wet steam region the calculation is carried out for saturated liquid and saturated steam only. The value -1000 will be shown for the wet steam region lying between.

Results for wrong input values

Result **ETA_PTXI_WALI = -1000**, **ETA = -1000** or **eta_ptxi_PROP = -1000** for input values:

p > 10 bar or p < 0.00074 bar or
t > 210 °C or t < 0 °C or
$\xi_s > \xi_{sol}(t)$ or

$\xi > 1.0$ kg/kg or $\xi < 0.3$ kg/kg
Points in the wet steam region between saturated liquid and saturated steam

References: [2], [22], [23]

Dynamic Viscosity of Saturated Liquid $\eta^l = f(p_s, t_s, \xi^l)$

Function Name:	**etal_pstsxil_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION ETAL_WALI(PS,TS,XIL)** REAL*8 PS,TS,XIL
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_ETAL_WALI(ETAL,PS,TS,XIL)** REAL*8 ETAL,PS,TS,XIL

Input Values:

PS - Vapor pressure p_s in bar

TS - Saturation temperature t_s in °C

XIL - Mass fraction of water as saturated liquid ξ^l in (kg H_2O)/(kg mixture)

Result

ETAL_WALI, **ETAL** or **etal_pstsxil_WaLi** - Dynamic viscosity of saturated liquid η^l in Pa s

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints of the input variants

For calculating the dynamic viscosity of the saturated liquid it is necessary to define two parameters (either t_s and ξ' or p_s and ξ' or p_s and t_s) and –1 must be entered as a pro-forma value for the value which is not given. When p_s, t_s, and ξ' are entered, the program considers p_s, t_s, and ξ' to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$$\eta^l = f(-1, t_s, \xi^l)$$

$$\eta^l = f(p_s, -1, \xi^l)$$

$$\eta^l = f(p_s, t_s, -1)$$

$$\eta^l = f(p_s, t_s, \xi^l)$$

Results for wrong input values

Result **ETAL_WALI = −1000**, **ETAL = −1000** or **etal_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

References: [2], [22], [23]

Dynamic Viscosity of Saturated Steam $\eta^{v} = f(p_s, t_s, \xi^{l})$

Function Name:	**etav_pstsxil_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION ETAV_WALI(PS,TS,XIL)** REAL*8 PS,TS,XIL
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_ETAV_WALI(ETAV,PS,TS,XIL)** REAL*8 ETAV,PS,TS,XIL

Input Values:

PS - Vapor pressure p_s in bar

TS - Saturation temperature t_s in °C

XIL - Mass fraction of water as saturated liquid ξ^{l} in (kg H_2O)/(kg mixture)

Result

ETAV_WALI, **ETAV** or **etav_pstsxil_WaLi** - Dynamic viscosity of saturated steam η^{v} in Pa s

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints of the input variants

For calculating the dynamic viscosity of saturated steam it is necessary to define two parameters (either t_s and ξ' or p_s and ξ' or p_s and t_s) and–1 must be entered as a pro-forma value for the value which is not given. When p_s, t_s, and ξ' are entered, the program considers p_s, t_s, and ξ' to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$$\eta^{v} = f(-1, t_s, \xi^{l})$$

$$\eta^{v} = f(p_s, -1, \xi^{l})$$

$$\eta^{v} = f(p_s, t_s, -1)$$

$$\eta^{v} = f(p_s, t_s, \xi^{l})$$

Results for wrong input values

Result **ETAV_WALI = −1000**, **ETAV = −1000** or **etav_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^{l} > 1.0$ kg/kg or $\xi^{l} < 0.3$ kg/kg

References: [2], [23]

Specific Enthalpy *h* = f(*p*,*t*,*ξ*)

Function Name:	**h_ptxi_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION H_PTXI_WALI(P,T,XI)** REAL*8 P,T,XI
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_H_PTXI_WALI(H,P,T,XI)** REAL*8 H,P,T,XI

Input Values:

P - Pressure p in bar

T - Temperature t in °C

XI - Mass fraction of water ξ in (kg H_2O)/(kg mixture)

Result

H_PTXI_WALI, **H** or **h_ptxi_WaLi** – Specific enthalpy h in kJ/kg

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints for wet steam

The wet steam region is calculated automatically by the subprograms:

Results for wrong input values

Result **H_PTXI_WALI = −1000**, **H = −1000** or **h_ptxi_WaLi = −1000** for input values:

p > 10 bar or p < 0.00074 bar or
t > 210 °C or t < 0 °C or
$\xi_s > \xi_{sol}(t)$ or

ξ > 1.0 kg/kg or ξ < 0.3 kg/kg

References: [2], [21], [23]

Specific Enthalpy of Saturated Liquid $h^l = f(p_s, t_s, \xi^l)$

Function Name:	**hl_pstsxil_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION HL_WALI(PS,TS,XIL)** REAL*8 PS,TS,XIL
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_HL_WALI(HL,PS,TS,XIL)** REAL*8 HL,PS,TS,XIL

Input Values:

PS - Vapor pressure p_s in bar

TS - Saturation temperature t_s in °C

XIL - Mass fraction of water as saturated liquid ξ^l in (kg H_2O)/(kg mixture)

Result

HL_WALI, **HL** or **hl_pstsxil_WaLi** - Specific enthalpy of saturated liquid h^l in kJ/kg

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints of the input variants

For calculating the specific enthalpy of the saturated liquid it is necessary to define two parameters (either t_S and ξ' or p_S and ξ' or p_S and t_S) and –1 must be entered as a pro-forma value for the value which is not given. When p_S, t_S, and ξ' are entered, the program considers p_S, t_S, and ξ' to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$h^l = f(-1, t_s, \xi^l)$

$h^l = f(p_s, -1, \xi^l)$

$h^l = f(p_s, t_s, -1)$

$h^l = f(p_s, t_s, \xi^l)$

Results for wrong input values

Result **HL_WALI = −1000**, **HL = −1000** or **hl_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

References: [2], [21], [23]

Specific Enthalpy of Saturated Steam $h^v = f(p_s, t_s, \xi^l)$

Function Name:	**hv_pstsxil_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION HV_WALI(PS,TS,XIL)** REAL*8 PS,TS,XIL
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_HV_WALI(HV,PS,TS,XIL)** REAL*8 HV,PS,TS,XIL

Input Values:

PS - Vapor pressure p_s in bar

TS - Saturation temperature t_s in °C

XIL - Mass fraction of water as saturated liquid ξ^l in (kg H_2O)/(kg mixture)

Result

HV _ WALI, **HV** or **hv _ pstsxil _ WaLi** - Specific enthalpy of saturated steam h^v in kJ/kg

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints of the input variants

For calculating the specific enthalpy of saturated steam it is necessary to define two parameters (either t_S and ξ' or p_S and ξ' or p_S and t_S) and –1 must be entered as a pro-forma value for the value which is not given. When p_S, t_S, and ξ' are entered, the program considers p_S, t_S, and ξ' to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$$h^v = f(-1, t_s, \xi^l)$$

$$h^v = f(p_s, -1, \xi^l)$$

$$h^v = f(p_s, t_s, -1)$$

$$h^v = f(p_s, t_s, \xi^l)$$

Results for wrong input values

Result **HV_WALI = −1000**, **HV = −1000** or **hv_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

References: [2], [23]

Specific Enthalpy at the Crystallization Barrier $h_{sol} = f(\xi)$

Function Name:	**hsol_xi_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION HSOL_XI_WALI(XI)** REAL*8 XI
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_HSOL_XI_WALI(HSOL,XI)** REAL*8 HSOL,XI

Input Values:

XI - Mass fraction of water ξ in (kg H_2O)/(kg mixture)

Result

HSOL_XI_WALI, **HSOL** or **hsol_xi_WaLi** - Specific enthalpy at the crystallization barrier h_{sol} in kJ/kg

Range of validity

Composition range: from 0.3 to 0.45 (kg H_2O)/(kg mixture)

Explanation of the function

This function calculates the specific enthalpy at the crystallization barrier. If the actual specific enthalpy calculated in the liquid phase of the mixture lies below the enthalpy calculated by the function then the Lithium Bromide crystallizes out. This case is recognised by the other functions. A calculation of properties below the crystallization barrier is not possible.

Results for wrong input values

Result **HSOL_XI_WALI = -1000**, **HSOL = -1000** or **hsol_xi_WaLi = -1000** for input values:

$\xi > 0.45$ kg/kg or $\xi < 0.3$ kg/kg

References: [23]

Thermal Conductivity $\lambda = f(p, t, \xi)$

Function Name:	**lam_ptxi_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION LAM_PTXI_WALI(P,T,XI)** REAL*8 P,T,XI
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_LAM_PTXI_WALI(LAM,P,T,XI)** REAL*8 LAM,P,T,XI

Input Values:

P - Pressure p in bar

T - Temperature t in °C

XI - Mass fraction of water ξ in (kg H_2O)/(kg mixture)

Result

LAM_PTXI_WALI, **LAM** or **lam_ptxi_WaLi** – Thermal conductivity λ in W/m K

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints for wet steam

In the wet steam region the calculation is carried out for saturated liquid and saturated steam only. The value -1000 will be shown for the wet steam region lying between.

Results for wrong input values

Result **LAM_PTXI_WALI = -1000**, **LAM = -1000** or **lam_ptxi_WaLi = -1000** for input values:

$p > 10$ bar or $p < 0.00074$ bar or
$t > 210$ °C or $t < 0$ °C oder
$\xi_s > \xi_{sol}(t)$ or
$\xi > 1.0$ kg/kg or $\xi < 0.3$ kg/kg
Points in the wet steam region between saturated liquid and saturated steam

References: [2], [22], [23]

Thermal Conductivity of Saturated Liquid $\lambda^l = f(p_s, t_s, \xi^l)$

Function Name:	**laml_pstsxil_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION LAML_WALI(PS,TS,XIL)** REAL*8 PS,TS,XIL
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_LAML_WALI(LAML,PS,TS,XIL)** REAL*8 ETAL,PS,TS,XIL

Input Values:

PS - Vapor pressure p_s in bar

TS - Saturation temperature t_s in °C

XIL - Mass fraction of water as saturated liquid ξ^l in (kg H_2O)/(kg mixture)

Result

LAML_WALI, **LAML** or **laml_pstsxil_WaLi** - Thermal conductivity of saturated liquid λ^l in W / (m K)

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints of the input variants

For calculating the thermal conductivity of the saturated liquid it is necessary to define two parameters (either t_S and ξ' or p_S and ξ' or p_S and t_S) and –1 must be entered as a pro-forma value for the value which is not given. When p_S, t_S, and ξ' are entered, the program considers p_S, t_S, and ξ' to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$$\lambda^l = f(-1, t_s, \xi^l)$$

$$\lambda^l = f(p_s, -1, \xi^l)$$

$$\lambda^l = f(p_s, t_s, -1)$$

$$\lambda^l = f(p_s, t_s, \xi^l)$$

Results for wrong input values

Result **LAML_WALI = −1000**, **LAML = −1000** or **laml_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

References: [2], [22], [23]

Thermal Conductivity of Saturated Steam $\lambda^v = f(p_s, t_s, \xi^l)$

Function Name:	**lamv_pstsxil_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION LAMV_WALI(PS,TS,XIL)** REAL*8 PS,TS,XIL
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_LAMV_WALI(LAMV,PS,TS,XIL)** REAL*8 LAMV,PS,TS,XIL

Input Values:

PS - Vapor pressure p_s in bar

TS - Saturation temperature t_s in °C

XIL - Mass fraction of water as saturated liquid ξ^l in (kg H_2O)/(kg mixture)

Result

LAMV_WALI, **LAMV** or **lamv_pstsxil_WaLi** - Thermal conductivity of saturated steam λ^v in W / (m K)

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints of the input variants

For calculating the thermal conductivity of saturated steam it is necessary to define two parameters (either t_S and ξ' or p_S and ξ' or p_S and t_S) and –1 must be entered as a pro-forma value for the value which is not given. When p_S, t_S, and ξ' are entered, the program considers p_S, t_S, and ξ' to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$$\lambda^v = f(-1, t_s, \xi^l)$$

$$\lambda^v = f(p_s, -1, \xi^l)$$

$$\lambda^v = f(p_s, t_s, -1)$$

$$\lambda^v = f(p_s, t_s, \xi^l)$$

Results for wrong input values

Result **LAMV_WALI = −1000**, **LAMV = −1000** or **lamv_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

References: [2], [23]

Kinematic Viscosity $\nu = f(p,t,\xi)$

Function Name:	**ny_ptxi_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION NY_PTXI_WALI(P,T,XI)** REAL*8 P,T,XI
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_NY_PTXI_WALI(NY,P,T,XI)** REAL*8 NY,P,T,XI

Input Values:

P - Pressure p in bar

T - Temperature t in °C

XI - Mass fraction of water ξ in (kg H_2O)/(kg mixture)

Result

NY_PTXI_WALI, **NY** or **ny_ptxi_WaLi** - Kinematic viscosity $\nu = \eta \cdot v$ in m^2/s

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints for wet steam

In the wet steam region the calculation is carried out for saturated liquid and saturated steam only. The value -1000 will be shown for the wet steam region lying between.

Results for wrong input values

Result **NY_PTXI_WALI = -1000**, **NY = -1000** or **ny_ptxi_WaLi = -1000** for input values:

$p > 10$ bar or $p < 0.00074$ bar or
$t > 210$ °C or $t < 0$ °C oder
$\xi_s > \xi_{sol}(t)$ or
$\xi > 1.0$ kg/kg or $\xi < 0.3$ kg/kg
Points in the wet steam region between saturated liquid and saturated steam

References: [2], [21], [22], [23]

Kinematic Viscosity of Saturated Liquid $\nu^l = f(p_s, t_s, \xi^l)$

Function Name:	**nyl_pstsxil_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION NYL_WALI(PS,TS,XIL)** REAL*8 PS,TS,XIL
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_NYL_WALI(NYL,PS,TS,XIL)** REAL*8 NYL,PS,TS,XIL

Input Values:

PS - Vapor pressure p_s in bar

TS - Saturation temperature t_s in °C

XIL - Mass fraction of water as saturated liquid ξ^l in (kg H_2O)/(kg mixture)

Result

NYL_WALI, **NYL** or **nyl_pstsxil_WaLi** - Kinematic viscosity of saturated liquid $\nu^l = \eta^l \cdot v^l$ in m^2/s

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints of the input variants

For calculating the kinematic viscosity of the saturated liquid it is necessary to define two parameters (either t_S and ξ' or p_S and ξ' or p_S and t_S) and –1 must be entered as a pro-forma value for the value which is not given. When p_S, t_S, and ξ' are entered, the program considers p_S, t_S, and ξ' to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$$\nu^l = f(-1, t_s, \xi^l)$$

$$\nu^l = f(p_s, -1, \xi^l)$$

$$\nu^l = f(p_s, t_s, -1)$$

$$\nu^l = f(p_s, t_s, \xi^l)$$

Results for wrong input values

Result **NYL_WALI = −1000**, **NYL = −1000** or **nyl_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

References: [2], [21], [22], [23]

Kinematic Viscosity of Saturated Steam $\nu^{v} = f(p_s, t_s, \xi^{l})$

Function Name:	**nyv_pstsxil_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION NYV_WALI(PS,TS,XIL)** REAL*8 PS,TS,XIL
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_NYV_WALI(NYV,PS,TS,XIL)** REAL*8 NYV,PS,TS,XIL

Input Values:

PS - Vapor pressure p_s in bar

TS - Saturation temperature t_s in °C

XIL - Mass fraction of water as saturated liquid ξ^{l} in (kg H_2O)/(kg mixture)

Result

NYV_WALI, **NYV** or **nyv_pstsxil_WaLi** - Kinematic viscosity of saturated steam $\nu^{v} = \eta^{v} \cdot v^{v}$ in m^2/s

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints of the input variants

For calculating the kinematic viscosity of the saturated liquid it is necessary to define two parameters (either t_S and ξ' or p_S and ξ' or p_S and t_S) and –1 must be entered as a pro-forma value for the value which is not given. When p_S, t_S, and ξ' are entered, the program considers p_S, t_S, and ξ' to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$$\nu^{v} = f(-1, t_s, \xi^{l})$$
$$\nu^{v} = f(p_s, -1, \xi^{l})$$
$$\nu^{v} = f(p_s, t_s, -1)$$
$$\nu^{v} = f(p_s, t_s, \xi^{l})$$

Results for wrong input values

Result **NYV_WALI = -1000**, **NYV = -1000** or **nyv_pstsxil_WaLi = -1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^{l} > 1.0$ kg/kg or $\xi^{l} < 0.3$ kg/kg

References: [2], [23]

Prandtl-Number $\mathbf{Pr = f(p,t,\xi)}$

Function Name:	**Pr_ptxi_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION PR_PTXI_WALI(P,T,XI)** REAL*8 P,T,XI
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_PR_PTXI_WALI(PR,P,T,XI)** REAL*8 PR,P,T,XI

Input Values:

P - Pressure p in bar

T - Temperature t in °C

XI - Mass fraction of water ξ in (kg H_2O)/(kg mixture)

Result

PR_PTXI_WALI, **PR** or **Pr_ptxi_WALI** - $\text{Prandtl-Number } \mathrm{Pr} = \frac{\eta \cdot c_p}{\lambda}$

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints for wet steam

In the wet steam region the calculation is carried out for saturated liquid and saturated steam only. The value -1000 will be shown for the wet steam region lying between.

Results for wrong input values

Result **PR_PTXI_WALI = -1000**, **PR = -1000** or **Pr_ptxi_WaLi = -1000** for input values:

$p > 10$ bar or $p < 0.00074$ bar or
$t > 210$ °C or $t < 0$ °C or
$\xi_s > \xi_{sol}(t)$ or
$\xi > 1.0$ kg/kg or $\xi < 0.3$ kg/kg
Points in the wet steam region between saturated liquid and saturated steam

References: [2], [21], [22], [23]

Prandtl-Number of Saturated Liquid $Pr^l = f(p_s, t_s, \xi^l)$

Function Name:	**Prl_pstsxil_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION PRL_WALI(PS,TS,XIL)** REAL*8 PS,TS,XIL
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_PRL_WALI(PRL,PS,TS,XIL)** REAL*8 PRL,PS,TS,XIL

Input Values:

PS - Vapor pressure p_s in bar

TS - Saturation temperature t_s in °C

XIL - Mass fraction of water as saturated liquid ξ^l in (kg H_2O)/(kg mixture)

Result

PRL_WALI, **PRL** or **Prl_pstsxil_WaLi** - Prandtl-Number of saturated liquid $Pr^l = \frac{\eta^l \cdot c_p{}^l}{\lambda^l}$

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints of the input variants

For calculating the Prandtl-Number of the saturated liquid it is necessary to define two parameters (either t_s and ξ' or p_s and ξ' or p_s and t_s) and –1 must be entered as a pro-forma value for the value which is not given. When p_s, t_s, and ξ' are entered, the program considers p_s, t_s, and ξ' to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$Pr^l = f(-1, t_s, \xi^l)$

$Pr^l = f(p_s, -1, \xi^l)$

$Pr^l = f(p_s, t_s, -1)$

$Pr^l = f(p_s, t_s, \xi^l)$

Results for wrong input values

Result **PRL_WALI = −1000**, **PRL = −1000** or **Prl_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

References: [2], [21], [22], [23]

Prandtl-Number of Saturated Steam $Pr^v = f(p_s, t_s, \xi^l)$

Function Name:	**Prv_pstsxil_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION PRV_WALI(PS,TS,XIL)** REAL*8 PS,TS,XIL
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_PRV_WALI(PRV,PS,TS,XIL)** REAL*8 PRV,PS,TS,XIL

Input Values:

PS - Vapor pressure p_s in bar

TS - Saturation temperature t_s in °C

XIL - Mass fraction of water as saturated liquid ξ^l in (kg H_2O)/(kg mixture)

Result

PRV_WALI, **PRV** or **Prv_pstsxil_WaLi** - Prandtl-Number of saturated steam $Pr^v = \dfrac{\eta^v \cdot c_p{}^v}{\lambda^v}$

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints of the input variants

For calculating the Prandtl-Number of saturated steam it is necessary to define two parameters (either t_S and ξ' or p_S and ξ' or p_S and t_S) and –1 must be entered as a pro-forma value for the value which is not given. When p_S, t_S, and ξ' are entered, the program considers p_S, t_S, and ξ' to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$$Pr^v = f(-1, t_s, \xi^l)$$

$$Pr^v = f(p_s, -1, \xi^l)$$

$$Pr^v = f(p_s, t_s, -1)$$

$$Pr^v = f(p_s, t_s, \xi^l)$$

Results for wrong input values

Result **PRV_WALI = −1000**, **PRV = −1000** or **Prv_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

References: [2], [23]

Vapor Pressure $p_s = f(t_s, \xi^l)$

Function Name:	**ps_tsxil_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION PS_TSXIL_WALI(TS,XIL)** REAL*8 TS,XIL
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_PS_TSXIL_WALI(PS,TS,XIL)** REAL*8 PS,TS,XIL

Input Values:

TS - Saturation temperature t_s in °C

XIL - Mass fraction of water as saturated liquid ξ^l in (kg H_2O)/(kg mixture)

Result

PS_TSXIL_WALI, PS or ps_tsxil_WaLi - Vapor pressure p_s in bar

Range of validity

Temperature range:	from 0 °C to 210 °C
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Results for wrong input values

Result **PS_TSXIL_WALI = −1000**, **PS = −1000** or **ps_tsxil_WALI = −1000** for input values:

$t_s > 210$ °C or $t_s < 0$ °C or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

$\xi^l < \xi_{sol}(t_s)$ or

References: [2], [21], [23]

Pressure at the Crystallization Barrier $p_{sol} = f(t)$

Function Name:	**psol_t_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION PSOL_T_WALI(T)** REAL*8 T
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_PSOL_T_WALI(PSOL,T)** REAL*8 PSOL,T

Input Values:

T - Temperature t in °C

Result

PSOL_T_WALI, **PSOL** or **psol_t_WaLi** - Pressure at the Crystallization barrier p_{sol} in bar

Range of validity

Temperature range: from 0 °C to 93.58 °C

Explanation of the function

This function calculates the pressure at the crystallization barrier. If the actual pressure calculated in the liquid phase of the mixture lies below the pressure calculated by the function then the Lithium Bromide crystallizes out. This case is recognised by the other functions. A calculation of properties below the crystallization barrier is not possible. Please note that this function puts out expedient results only if a liquid phase is calculated since there is pure water in the gas phase.

Results for wrong input values

Result **PSOL_T_WALI = −1000**, **PSOL = −1000** or **psol_t_WaLi = −1000** for input values:

t > 93.58 °C or t < 0 °C

References: [2], [21], [23]

Phase Region Region = f(p,t,ξ)

Function Name:	**region_ptxi_WaLi**
Subroutine with function value: for call from Fortran	**INTEGER*4 FUNCTION REGION_PTXI_WALI(P,T,XI)** REAL*8 P,T,XI
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_REGION_PTXI_WALI(REGION,P,T,XI)** INTEGER*4 REGION REAL*8 P,T,XI

Input Values:

P	-	Pressure p in bar
T	-	Temperature t in °C
XI	-	Mass fraction of water ξ in (kg H_2O)/(kg mixture)

Result

REGION_PTXI_WALI, **REGION** or **region_ptxi_WaLi** – Phase region

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Explanation of the function

This function defines the phase region at hand. The results have the following meaning:

1 – Subcooled fluid
2 – Two phase region
3 – Gas phase

If a point of state is entered below the crystallization barrier the result will be – 2000.

Results for wrong input values

Result **REGION_PTXI_WALI = −1000**, **REGION = −1000** or **region_ptxi_WaLi = −1000** for input values:

$p > 10$ bar or $p < 0.00074$ bar or
$t > 210$ °C or $t < 0$ °C oder
$\xi > 1.0$ kg/kg or $\xi < 0.3$ kg/kg

Result **REGION_PTXI_WALI = −2000, REGION = −2000** or **region_ptxi_WaLi = −2000** for input values:

$\xi < \xi_{sol}(t)$

References: [2], [21], [23]

Phase Region Region = f(p,h,ξ)

Function Name:	**region_phxi_WaLi**
Subroutine with function value: for call from Fortran	**INTEGER*4 FUNCTION REGION_PHXI_WALI(P,H,XI)** REAL*8 P,H,XI
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_REGION_PHXI_WALI(REGION,P,H,XI)** INTEGER*4 REGION REAL*8 P,H,XI

Input Values:

P - Pressure p in bar

H - Specific enthalpy h in kJ/kg

XI - Mass fraction of water ξ in (kg H_2O)/(kg mixture)

Result

REGION_PHXI_WALI, **REGION** or **region_phxi_WaLi** – Phase region

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Explanation of the function

This function defines the phase region at hand. The results have the following meaning:

1 – Subcooled fluid
2 – Two phase region
3 – Gas phase

If a point of state is entered below the crystallization barrier the result will be – 2000.

Results for wrong input values

Result **REGION_PHXI_WALI = −1000**, **REGION = −1000** or **region_phxi_WaLi = −1000** for input values:

$p > 10$ bar or $p < 0.00074$ bar or
$h > 2899.46775$ kJ/kg or $h < 14$ kJ/kg or
$\xi > 1.0$ kg/kg or $\xi < 0.3$ kg/kg

Result **REGION_PHXI_WALI = -2000, REGION = -2000** or **region_phxi_WaLi = -2000** for input values:

$h < h_{sol}(\xi)$

References: [2], [21], [23]

Phase Region Region = f(p,s,ξ)

Function Name:	**region_psxi_WaLi**
Subroutine with function value: for call from Fortran	**INTEGER*4 FUNCTION REGION_PSXI_WALI(P,S,XI)** REAL*8 P,H,XI
Subroutine with parameter: für Aufruf aus DLL	**INTEGER*4 FUNCTION C_REGION_PSXI_WALI(REGION,P,S,XI)** INTEGER*4 REGION REAL*8 P,S,XI

Input Values:

P - Pressure p in bar

S - Specific entropy s in kJ/(kg K)

XI - Mass fraction of water ξ in (kg H_2O)/(kg mixture)

Result

REGION_PSXI_WALI, **REGION** or **region_psxi_WaLi** – Phase region

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Explanation of the function

This function defines the phase region at hand. The results have the following meaning:

1 – Subcooled fluid
2 – Two phase region
3 – Gas phase

If a point of state is entered below the crystallization barrier the result will be – 2000.

Results for wrong input values

Result **REGION_PHXI_WALI = −1000**, **REGION = −1000** or **region_phxi_WaLi = −1000** for input values:

p > 10 bar or p < 0.00074 bar or
at intermediate result t > 210 °C or t < 0 °C or
$\xi > 1.0$ kg/kg or $\xi < 0.3$ kg/kg

Result **REGION_PSXI_WALI = -2000, REGION = -2000** or **region_psxi_WaLi = -2000** for input values:

$\xi < \xi_{sol}(p)$

References: [2], [21], [23]

Specific Entropy s = f(p,t,ξ)

Function Name:	**s_ptxi_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION S_PTXI_WALI(P,T,XI)** REAL*8 P,T,X
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_S_PTXI_WALI(S,P,T,XI)** REAL*8 S,P,T,XI

Input Values:

P - Pressure p in bar

T - Temperature t in °C

XI - Mass fraction from water ξ in (kg H_2O)/(kg mixture)

Result

S_PTXI_WALI, **S** or **s_ptxi_WaLi** - Specific entropy s in kJ/kg K

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints for wet steam

The wet steam region is calculated automatically by the subprograms:

Results for wrong input values

Result **S_PTXI_WALI = −1000**, **S = −1000** or **s_ptxi_WaLi = −1000** for input values:

p > 10 bar or p < 0.00074 bar or
t > 210 °C or t < 0 °C oder
$\xi < \xi_{sol}(t)$
$\xi > 1.0$ kg/kg or $\xi < 0.3$ kg/kg

References: [2], [21], [23]

Specific Entropy of Saturated Liquid $s^l = f(p_s, t_s, \xi^l)$

Function Name:	**sl_pstsxil_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION SL_WALI(PS,TS,XIL)** REAL*8 PS,TS,XIL
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_SL_WALI(HL,PS,TS,XIL)** REAL*8 SL,PS,TS,XIL

Input Values:

PS - Vapor pressure p_s in bar

TS - Saturation temperature t_s in °C

XIL - Mass fraction of water as saturated liquid ξ^l in (kg H_2O)/(kg mixture)

Result

SL_WALI, **SL** or **sl_pstsxil_WaLi** - Specific entropy of the saturated liquid s^l in kJ / (kg K)

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints of the input variants

For calculating the specific entropy of the saturated liquid it is necessary to define two parameters (either t_S and ξ' or p_S and ξ' or p_S and t_S) and –1 must be entered as a pro-forma value for the value which is not given. When p_S, t_S, and ξ' are entered, the program considers p_S, t_S, and ξ' to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$$s^l = f(-1, t_s, \xi^l)$$

$$s^l = f(p_s, -1, \xi^l)$$

$$s^l = f(p_s, t_s, -1)$$

$$s^l = f(p_s, t_s, \xi^l)$$

Results for wrong input values

Result **SL_WALI = −1000**, **SL = −1000** or **sl_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

References: [2], [21], [23]

Specific Entropy of Saturated Steam $s^v = f(p_s, t_s, \xi^l)$

Function Name:	**sv_pstsxil_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION SV_WALI(PS,TS,XIL)** REAL*8 PS,TS,XIL
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_SV_WALI(SV,PS,TS,XIL)** REAL*8 SV,PS,TS,XIL

Input Values:

PS - Vapor pressure p_s in bar

TS - Saturation temperature t_s in °C

XIL - Mass fraction of water as saturated liquid ξ^l in (kg H_2O)/(kg mixture)

Result

SV_WALI, **SV** or **sv_pstsxil_WaLi** - Specific entropy of saturated steam s^v in kJ/(kg K)

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints of the input variants

For calculating the specific entropy of saturated steam it is necessary to define two parameters (either t_S and ξ' or p_S and ξ' or p_S and t_S) and –1 must be entered as a pro-forma value for the value which is not given. When p_S, t_S, and ξ' are entered, the program considers p_S, t_S, and ξ' to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$s^v = f(-1, t_s, \xi^l)$

$s^v = f(p_s, -1, \xi^l)$

$s^v = f(p_s, t_s, -1)$

$s^v = f(p_s, t_s, \xi^l)$

Results for wrong input values

Result **SV_WALI = −1000**, **SV = −1000** or **sv_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

References: [2], [23]

Backward Function: Temperature t = f(p,h,ξ)

Function Name:	**t_phxi_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION T_PHXI_WALI(P,H,XI)** REAL*8 P,H,XI
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_T_PHXI_WALI(T,P,H,XI)** REAL*8 T,P,H,XI

Input Values:

P	-	Pressure p in bar
H	-	Specific enthalpy h in kJ/kg
XI	-	Mass fraction of water ξ in (kg H_2O)/(kg mixture)

Result

T_PHXI_WALI, **T** or **t_phxi_WaLi** – Temperature t in °C

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints for wet steam

The wet steam region is calculated automatically by the subprograms:

Results for wrong input values

Result **T_PHXI_WALI = −1000**, **T = −1000** or **t_phxi_WaLi = −1000** for input values:

p > 10 bar or p < 0.00074 bar or
at calculation result t > 210 °C or t < 0 °C or
ξ > 1.0 kg/kg or ξ < 0.3 kg/kg or
h < h_{sol}(ξ)

References: [2], [21], [23]

Backward Function: Temperature t = f(p,s,ξ)

Function Name:	**t_psxi_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION T_PSXI_WALI(P,S,XI)** REAL*8 P,S,XI
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_T_PSXI_WALI(T,P,S,XI)** REAL*8 T,P,S,XI

Input Values:

P	-	Pressure p in bar
S	-	Specific entropy s in kJ/(kg K)
XI	-	Mass fraction of water ξ in (kg H_2O)/(kg mixture)

Result

T_PSXI_WALI, **T** or **t_psxi_WaLi** – Temperature *t* in °C

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints for wet steam

The wet steam region is calculated automatically by the subprograms:

Results for wrong input values

Result **T_PSXI_WALI = −1000**, **T = −1000** or **t_psxi_WaLi = −1000** for input values:

p > 10 bar or p < 0.00074 bar or
at calculation result t > 210 °C oder t < 0 °C oder
$\xi < \xi_{sol}(t)$
ξ > 1.0 kg/kg or ξ < 0.3 kg/kg

References: [2], [21], [23]

Saturation Temperature $t_s = f(p_s, \xi^l)$

Function Name: **ts_psxil_WaLi**

Subroutine with function value:
for call from Fortran

REAL*8 FUNCTION TS_PSXIL_WALI(PS,XIL)
REAL*8 P,XIL

Subroutine with parameter:
for call from DLL

INTEGER*4 FUNCTION C_TS_PSXIL_WALI(TS,PS,XIL)
REAL*8 TS,PS,XIL

Input Values:

PS - Vapor pressure p_s in bar

XIL - Mass fraction of water as saturated liquid ξ^l in (kg H_2O)/(kg mixture)

Result

TS_PSXIL_WALI, **TS** or **ts_psxil_WaLi** - Saturation temperature t_s in °C

Range of validity

Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Results for wrong input values

Result **TS_PSXIL_WALI = −1000**, **TS = −1000** or **ts_psxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

at calculation result $t_s > 210$ °C or $t_s < 0$ °C or

$\xi < \xi_{sol}(t)$

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

References: [2], [21], [23]

Temperature at the Crystallization Barrier $t_{sol} = f(p)$

Function Name:	**tsol_p_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION TSOL_P_WALI(P)** REAL*8 P
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_TSOL_P_WALI(TSOL,P)** REAL*8 TSOL,P

Input Values:

P - Pressure p in bar

Result

TSOL _ P _ WALI, **TSOL** or **tsol _ p _ WaLi** - Temperature at the crystallization barrier t_{sol} in °C

Range of validity

Pressure range: from 0.00074 bar to 0.0373 bar

Explanation of the function

This function calculates the temperature at the crystallization barrier. If the actual temperature calculated in the liquid phase of the mixture lies above the temperature calculated by the function then the Lithium Bromide crystallizes out. This case is recognised by the other functions. A calculation of properties below the crystallization barrier is not possible. Please note that this function puts out expedient results only if a liquid phase is calculated since there is pure water in the gas phase.

Results for wrong input values

Result **TSOL_P_WALI, = −1000**, **TSOL = −1000** or **tsol_p_WaLi = −1000** for input values:

p < 0.0373 bar or p > 0.00074 bar

References: [2], [21], [23]

Specific Volume $v = f(p,t,\xi)$

Function Name:	**v_ptxi_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION V_PTXI_WALI(P,T,XI)** REAL*8 P,T,XI
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_V_PTXI_WALI(V,P,T,XI)** REAL*8 V,P,T,XI

Input Values:

P	-	Pressure p in bar
T	-	Temperature t in °C
XI	-	Mass fraction of water ξ in (kg H_2O)/(kg mixture)

Result

V _PTXI _ WALI, **V** or **v _ptxi _ WaLi** - Specific volume v in m^3/kg

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints for wet steam

The wet steam region is calculated automatically by the subprograms:

Results for wrong input values

Result **V_PTXI_WALI = −1000**, **V = −1000** or **v_ptxi_WaLi = −1000** for input values:

$p > 10$ bar or $p < 0.00074$ bar or
$t > 210$ °C or $t < 0$ °C oder
$\xi < \xi_{sol}(t)$
$\xi > 1.0$ kg/kg or $\xi < 0.3$ kg/kg

References: [2], [21], [23]

Specific Volume of Saturated Liquid $v^l = f(p_s, t_s, \xi^l)$

Function Name:	**vl_pstsxil_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION VL_WALI(PS,TS,XIL)** REAL*8 PS,TS,XIL
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_VL_WALI(VL,PS,TS,XIL)** REAL*8 VL,PS,TS,XIL

Input Values:

PS - Vapor pressure p_s in bar

TS - Saturation temperature t_s in °C

XIL - Mass fraction of water as saturated liquid ξ^l in (kg H_2O)/(kg mixture)

Result

VL_WALI, **VL** or **vl_pstsxil_WaLi** - Specific volume of saturated liquid v^l in m^3/kg

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints of the input variants

For calculating the specific volume of the saturated liquid it is necessary to define two parameters (either t_S and ξ' or p_S and ξ' or p_S and t_S) and –1 must be entered as a pro-forma value for the value which is not given. When p_S, t_S, and ξ' are entered, the program considers p_S, t_S, and ξ' to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$$v^l = f(-1, t_s, \xi^l)$$

$$v^l = f(p_s, -1, \xi^l)$$

$$v^l = f(p_s, t_s, -1)$$

$$v^l = f(p_s, t_s, \xi^l)$$

Results for wrong input values

Result **VL_WALI = −1000**, **VL = −1000** or **vl_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

References: [2], [21], [23]

Specific Volume of Saturated Steam $v^v = f(p_s, t_s, \xi^l)$

Function Name:	**vv_pstsxil_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION VV_WALI(PS,TS,XIL)** REAL*8 PS,TS,XIL
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_VV_WALI(VV,PS,TS,XIL)** REAL*8 VV,PS,TS,XIL

Input Values:

PS - Vapor pressure p_s in bar

TS - Saturation temperature t_s in °C

XIL - Mass fraction of water as saturated liquid ξ^l in (kg H_2O)/(kg mixture)

Result

VV _ WALI, **VV** or **vv _ pstsxil _ WaLi** - Specific volume of saturated steam v^v in m^3/kg

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar
Composition range:	from 0.3 to 1.0 (kg H_2O)/(kg mixture)

Hints of the input variants

For calculating the specific volume of saturated steam it is necessary to define two parameters (either t_s and ξ' or p_s and ξ' or p_s and t_s) and –1 must be entered as a pro-forma value for the value which is not given. When p_s, t_s, and ξ' are entered, the program considers p_s, t_s, and ξ' to be appropriate to represent the vapor pressure curve. If it is not the case the calculation for the quantity of the chosen function to be calculated results in –1000.

Possible input variants:

$$v^v = f(-1, t_s, \xi^l)$$

$$v^v = f(p_s, -1, \xi^l)$$

$$v^v = f(p_s, t_s, -1)$$

$$v^v = f(p_s, t_s, \xi^l)$$

Results for wrong input values

Result **VV_WALI = −1000**, **VV = −1000** or **vv_pstsxil_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

$\xi^l > 1.0$ kg/kg or $\xi^l < 0.3$ kg/kg

References: [2], [21], [23]

Mass Fraction H_2O of Saturated Liquid $\xi^l = f(p_s, t_s)$

Function Name:	**xil_psts_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION XIL_PSTS_WALI(PS,TS)** REAL*8 PS,TS
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_XIL_PSTS_WALI(XIL,PS,TS)** REAL*8 XIL,PS,TS

Input Values:

PS - Vapor pressure p_s in bar

TS - Saturation temperature t_s in °C

Result

XIL_PSTS_WALI, **XIL** or **xil_psts_WaLi** - Mass fraction H_2O of saturated liquid ξ^l in kg H_2O / kg

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar

Results for wrong input values

Result **XIL_PSTS_WALI = −1000**, **XIL = −1000** or **xil_psts_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or

$t_s > 210$ °C or $t_s < 0$ °C or

$t_s > t_{sol}(p_s)$ or

References: [2], [21], [23]

Mass Fraction H_2O at the Crystallization Barrier $\xi_{sol} = f(p)$

Function Name:	**xisol_p_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION XISOL_P_WALI(P)** REAL*8 P
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_XISOL_P_WALI(XISOL,P)** REAL*8 XISOL,P

Input Values:

P - Pressure p in bar

Result

XISOL _P _WALI, **XISOL** or **xisol _p _WaLi** - Mass fraction H_2O at the crystallization barrier ξ_{sol} in kg H_2O /kg

Range of validity

Pressure range: from 0.00074 bar to 0.0373 bar

Explanation of the function

This function calculates the mass fraction H_2O at the crystallization barrier. If the actual mass fraction H_2O calculated in the liquid phase of the mixture lies below the mass fraction calculated by the function then the Lithium Bromide crystallizes out. This case is recognised by the other functions. A calculation of properties below the crystallization barrier is not possible.

Results for wrong input values

Result **XISOL_P_WALI, = −1000**, **XISOL = −1000** or **xisol_p_WaLi = −1000** for input values:

$p < 0.0373$ bar or $p > 0.00074$ bar

References: [2], [21], [23]

Mass Fraction H_2O at the Crystallization Barrier $\xi_{sol} = f(t)$

Function Name:	**xisol_t_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION XISOL_T_WALI(T)** REAL*8 T
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_XISOL_T_WALI(XISOL,T)** REAL*8 XISOL,T

Input Values:

T - Temperature t in °C

Result

XISOL _ T _ WALI, **XISOL** or **xisol _ t _ WaLi** - Mass fraction H_2O at the crystallization barrier ξ_{sol} in kg H_2O /kg

Range of validity

Temperature range: from 0 °C to 93.58 °C

Explanation of the function

This function calculates the mass fraction H_2O at the crystallization barrier. If the actual mass fraction H_2O calculated in the liquid phase of the mixture lies below the mass fraction calculated by the function then the Lithium Bromide crystallizes out. This case is recognised by the other functions. A calculation of properties below the crystallization barrier is not possible.

Results for wrong input values

Result **XISOL_T_WALI = −1000**, **XISOL = −1000** or **xisol_t_WaLi = −1000** for input values:

t > 93.58 °C or t < 0 °C

References: [2], [21], [23]

Mass Fraction H_2O of Saturated Steam $\xi^{v} = f(p_s, t_s)$

Function Name:	**xiv_psts_WaLi**
Subroutine with function value: for call from Fortran	**REAL*8 FUNCTION XIV_PSTS_WALI(PS,TS)** REAL*8 PS,TS
Subroutine with parameter: for call from DLL	**INTEGER*4 FUNCTION C_XIV_PSTS_WALI(XIV,PS,TS)** REAL*8 XIV,PS,TS

Input Values:

PS - Vapor pressure p_s in bar

TS - Saturation temperature t_s in °C

Result

XIV _PSTS _ WALI, **XIV** or **xiv _psts _ WaLi** - Mass fraction H_2O of saturated steam ξ^{v} in kg H_2O / kg

Range of validity

Temperature range:	from 0 °C to 210 °C
Pressure range:	from 0.00074 bar to 10 bar

Explanation of the function

This function calculates the mass fraction of water in the gas phase. Since only very few Lithium Bromide enters the gas phase, the program library acts on the assumption that there is only water in the gas phase. This is also the case for all other functions. For there is only water in the saturated steam, the function will always calculate a result of – 1 within the range of validity.

Results for wrong input values

Result **XIV_PSTS_WALI = −1000**, **XIV = −1000** or **xiv_psts_WaLi = −1000** for input values:

$p_s > 10$ bar or $p_s < 0.00074$ bar or
$t_s > 210$ °C or $t_s < 0$ °C or

References: [2], [21]

4. Property Libraries for Calculating Heat Cycles, Boilers, Turbines, and Refrigerators

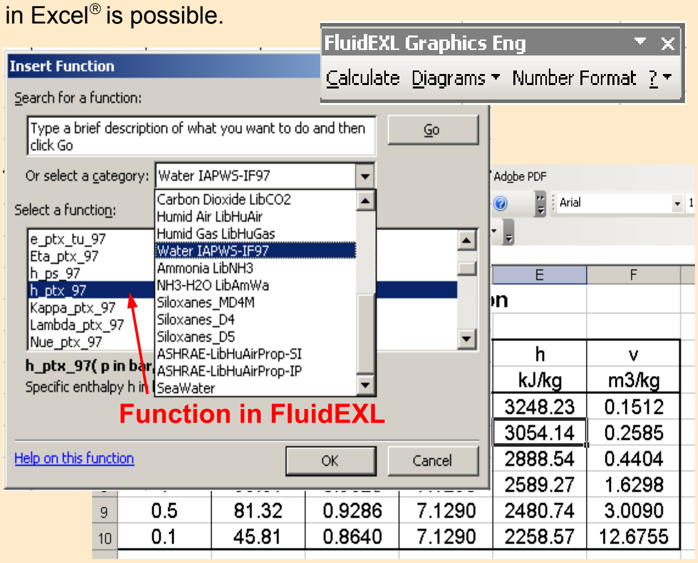
Water and Steam	Humid Combustion Gas Mixtures	Humid Air
Library LibIF97 - Industrial Formulation IAPWS-IF97 (Revision 2007) - Supplementary Standards - IAPWS-IF97-S01 - IAPWS-IF97-S03rev - IAPWS-IF97-S04 - IAPWS-IF97-S05 - IAPWS Revised Advisory Note No. 3 on Thermodynamic Derivatives (2008)	**Library LibHuGas** Ideal mixture of the real fluids: CO_2 - Span and Wagner H_2O - IAPWS-95 N_2 - Span et al. O_2 - Schmidt and Wagner Ar - Tegeler et al. and of the ideal gases: SO_2, CO, Ne (scientific equations of Bücker et al.) Consideration of Dissociation from VDI 4670 and Poynting effect	**Library LibHuAir** Ideal mixture of the real fluids: - Dry air from Lemmon et al. - Steam and water from IAPWS-IF97 Consideration of - Dissociation from VDI-Guideline 4670 - Poynting effect
	Library LibIDGAS Ideal gas mixture calculated from the VDI-Guideline 4670	**Library LibIdAir** Ideal gas mixture calculated from VDI-Guideline 4670

Carbon Dioxide	Ideal Gas Mixtures	Seawater
Library LibCO2 Formulation of Span and Wagner (1994)	**Library LibIdGasMix** Ideal mixture of the ideal gases: Ar, SO_2, Methane Ne, H_2, Ethane N_2, H_2S, Ethylene O_2, OH, Propylene CO, He, Propane CO_2, F_2, Iso-Butane Air, NH_3, n-Butane NO, Benzene H_2O, Methanol Consideration of - Dissociation from VDI-Guideline 4670	**Library LibSeaWa** IAPWS Formulation (2008) and IAPWS-IF97
Hydrogen		**Refrigerant R134a**
Library LibH2 Formulation of Leachman et al. (2007)		**Library LibR134a** Formulation of Tillner-Roth and Baehr (1994)
Helium		**Refrigerant NH_3**
Library LibHe Formulation of McCarty and Arp (1990)		**Library LibNH3** Formulation of Tillner-Roth (1995)
Methanol		
Library LibCH3OH Formulation of de Reuck and Craven (1993)		

ORC Working Fluids	Mixtures for Absorption Processes	Refrigerants
Library LibMM **Siloxane $C_6H_{18}OSi_2$ (MM)** Formulation of Colonna et al. (2006) **Library LibD4** **Siloxane $C_8H_{24}O_4Si_4$ (D4)** Formulation of Colonna et al. (2006) **Library LibD5** **Siloxane $C_{10}H_{30}O_5Si_5$ (D5)** Formulation of Colonna et al. (2006) **Library LibMD4M** **Siloxane $C_{14}H_{42}O_5Si_6$ (MD4M)** Formulation of Colonna et al. (2006)	**Library LibAmWa** **Ammonia/Water Mixtures** IAPWS Guideline 2005 of Tillner-Roth and Friend (1998) Helmholtz energy formulation for the mixing term **Library LibWaLi** **Water/Lithium Bromide Mixtures** Formulation of Kim and Infante Ferreira (2004) Gibbs energy equation for the mixing term	**Library LibPropan** **Refrigerant Propane** Formulation of Lemmon et al. (2008) **Library LibButan_Iso** **Refrigerant Iso-Butane** Formulation of Bücker et al. (2003) **Library LibButan_n** **Refrigerant n-Butane** Formulation of Bücker et al. (2003)

Add-In FluidEXL for Excel®

Using the Add-In FluidEXL a direct call of the property functions in Excel® is possible.



FluidEXL Graphics Eng

Insert Function

Search for a function: Type a brief description of what you want to do and then click Go

Or select a category: Water IAPWS-IF97

Carbon Dioxide LibCO2
Humid Air LibHuAir
Humid Gas LibHuGas
Water IAPWS-IF97
Ammonia LibNH3
NH3-H2O LibAmWa
Siloxanes_MD4M
Siloxanes_D4
Siloxanes_D5
ASHRAE-LibHuAirProp-SI
ASHRAE-LibHuAirProp-IP
SeaWater

Select a function:

e_ptx_tu_97
Eta_ptx_97
h_ps_97
h_ptx_97
Kappa_ptx_97
Lambda_ptx_97
Nue_ptx_97

h_ptx_97(p in bar,
Specific enthalpy h in

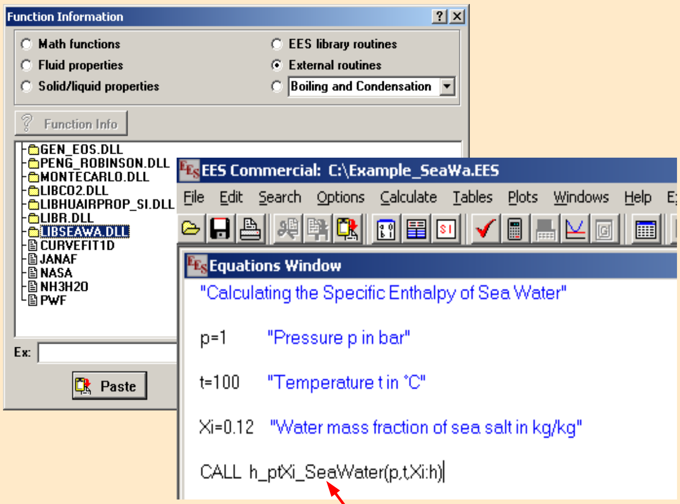
Help on this function

Function in FluidEXL

					h kJ/kg	v m3/kg
					3248.23	0.1512
					3054.14	0.2585
					2888.54	0.4404
					2589.27	1.6298
9	0.5	81.32	0.9286	7.1290	2480.74	3.0090
10	0.1	45.81	0.8640	7.1290	2258.57	12.6755

Add-In FluidEES for EES®

The Add-In FluidEES allows to call the functions of the property libraries within the Engineering Equation Solver EES®.



Function Information

○ Math functions
○ Fluid properties
○ Solid/liquid properties
○ EES library routines
● External routines
○ Boiling and Condensation

GEN_EOS.DLL
PENG_ROBINSON.DLL
MONTECARLO.DLL
LIBCO2.DLL
LIBHUAIRPROP_SI.DLL
LIBR.DLL
LIBSEAWA.DLL
CURVEFIT1D
JANAF
NASA
NH3H20
PWF

EES Commercial: C:\Example_SeaWa.EES

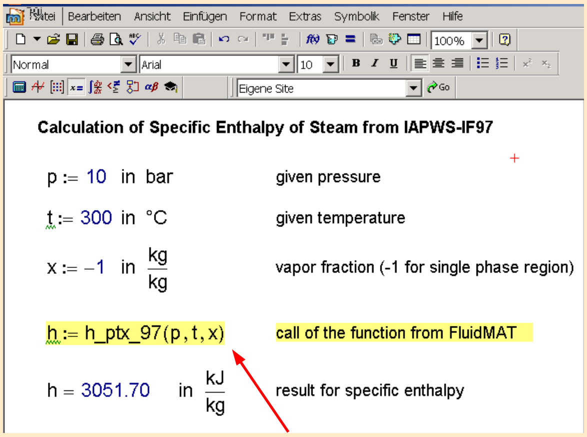
Equations Window

```
"Calculating the Specific Enthalpy of Sea Water"

p=1        "Pressure p in bar"

t=100      "Temperature t in °C"

Xi=0.12    "Water mass fraction of sea salt in kg/kg"

CALL h_ptXi_SeaWater(p,t,Xi:h)
```

Function in FluidEES

Add-In FluidMAT for Mathcad®

Using the Add-on FluidMAT, the functions of the property libraries can be used in Mathcad®.



Calculation of Specific Enthalpy of Steam from IAPWS-IF97

$p := 10$ in bar — given pressure

$t := 300$ in °C — given temperature

$x := -1$ in $\frac{kg}{kg}$ — vapor fraction (-1 for single phase region)

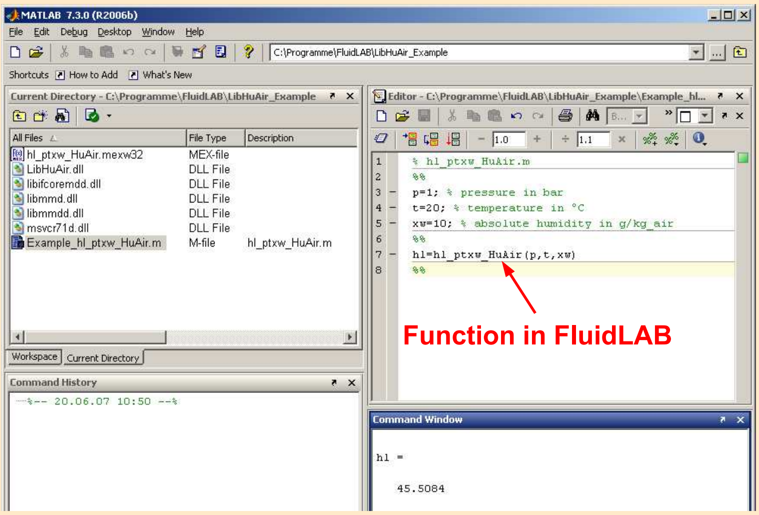
$h := h\_ptx\_97(p,t,x)$ — call of the function from FluidMAT

$h = 3051.70$ in $\frac{kJ}{kg}$ — result for specific enthalpy

Function in FluidMAT

Add-In FluidLAB for MATLAB®

The property functions can be called in MATLAB®.



MATLAB 7.3.0 (R2006b)

C:\Programme\FluidLAB\LibHuAir_Example

Current Directory - C:\Programme\FluidLAB\LibHuAir_Example

All Files	File Type	Description
hl_ptxw_HuAir.mexw32	MEX-file	
LibHuAir.dll	DLL File	
libifcoremdd.dll	DLL File	
libmmd.dll	DLL File	
libmmdd.dll	DLL File	
msvcr71d.dll	DLL File	
Example_hl_ptxw_HuAir.m	M-file	hl_ptxw_HuAir.m

Editor - C:\Programme\FluidLAB\LibHuAir_Example\Example_hl...

```
% hl_ptxw_HuAir.m
%%
p=1; % pressure in bar
t=20; % temperature in °C
xw=10; % absolute humidity in g/kg_air
%%
hl=hl_ptxw_HuAir(p,t,xw)
%%
```

Command History

```
%-- 20.06.07 10:50 --%
```

Command Window

```
hl =

   45.5084
```

Function in FluidLAB

The following thermodynamic and transport properties can be calculated[1]:

Thermodynamic Properties

- Saturation pressure p_s
- Saturation temperature T_s
- Density ρ
- Specific volume v
- Enthalpy h
- Internal energy u
- Entropy s
- Exergy e
- Isobaric heat capacity c_p
- Isochoric heat capacity c_v
- Isentropic exponent κ
- Speed of sound w
- Surface tension σ

Thermodynamic Derivatives

- Partial derivatives can be calculated

Transport Properties

- Dynamic viscosity η
- Kinematic viscosity ν
- Thermal conductivity λ
- *Prandtl*-number *Pr*

Backward Functions

- T, v, s (p,h)
- T, v, h (p,s)
- p, T, v (h,s)
- p, T (v,h)
- p, T (v,u)

For information please contact:

Zittau/Goerlitz University of Applied Sciences
Department of Technical Thermodynamics
Prof. Hans-Joachim Kretzschmar
Dr. Ines Stoecker
Theodor-Koerner-Allee 16
02763 Zittau, Germany

E-mail: hj.kretzschmar@hs-zigr.de
Internet: www.thermodynamics-zittau.de
Phone: +49-3583-61-1846
Fax.: +49-3583-61-1846

[1] Not all of these property functions are available in all property libraries listed before.

5 References

[1] Release on the IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam IAPWS-IF97.
IAPWS Sekretariat, Dooley, B, EPRI, Palo Alto CA (1997)

[2] Wagner, W.; Kruse, A.:
Zustandsgrößen von Wasser und Wasserdampf.
Springer-Verlag, Berlin (1998)

[3] Wagner, W.; Cooper, J.R.; Dittmann, A.; Kijima, J.; Kretzschmar, H.-J.; Kruse, A.; Mareš, R.; Oguchi, K.; Sato, H.; Stöcker, I.; Šifner, O.; Takaishi, Y.; Tanishita, I.; Trübenbach, J.; Willkommen, Th.:
The IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam.
ASME Journal of Eng. for Gas Turbines and Power 122 (2000) Nr. 1, S. 150-182

[4] Kretzschmar, H.-J.; Stöcker, I.; Klinger, J.; Dittmann, A.:
Calculation of Thermodynamic Derivatives for Water and Steam Using the New Industrial Formulation IAPWS-IF97.
in: Steam, Water and Hydrothermal Systems: Physics and Chemistry Meeting the Needs of Industry, Proceedings of the 13th International Conference on the Properties of Water and Steam, Eds. P.G. Hill et al., NRC Press, Ottawa, 2000

[5] Kretzschmar, H.-J.:
Mollier h,s-Diagramm. Springer-Verlag, Berlin (1998)

[6] Revised Release on the IAPS Formulation 1985 for the Thermal Conductivity of Ordinary Water Substance.
IAPWS Sekretariat, Dooley, B., EPRI, Palo Alto CA, (1997)

[7] Revised Release on the IAPS Formulation 1985 for the Viscosity of Ordinary Water Substance.
IAPWS Secretariat, Dooley, B., EPRI, Palo Alto CA, (1997)

[8] IAPWS Release on Surface Tension of Ordinary Water Substance 1994.
IAPWS Sekretariat, Dooley, B., EPRI, Palo Alto CA, (1994)

[9] Kretzschmar, H.-J.; Stöcker, I.; Willkommen, Th.; Trübenbach, J.; Dittmann, A.:
Supplementary Equations $v(p,T)$ for the Critical Region to the New Industrial Formulation IAPWS-IF97 for Water and Steam.
in: Steam, Water and Hydrothermal Systems: Physics and ChemistryMeeting the Needs of Industry, Proceedings of the 13th International Conference on the Properties of Water and Steam, Eds. P.G. Hill et al., NRC Press, Ottawa, 2000

[10] Kretzschmar, H.-J.; Cooper, J.R.; Dittmann, A.; Friend, D.G.; Gallagher, J.; Knobloch, K.; Mareš, R.; Miyagawa, K.; Stöcker, I.; Trübenbach, J.; Willkommen, Th.:
Supplementary Backward Equations for Pressure as a Function of Enthalpy and Entropy *p*(*h*,*s*) to the Industrial Formulation IAPWS-IF97 for Water and Steam.
ASME Journal of Engineering for Gas Turbines and Power – in preparation

[11] Release on the IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use.
IAPWS Sekretariat, Dooley, B., EPRI, Palo Alto CA, (1995)

[12] Grigull, U.:
Properties of Water and Steam in SI Units.
Springer-Verlag, Berlin (1989)

[13] Kretzschmar, H.-J.:
Zur Aufbereitung und Darbietung thermophysikalischer Stoffdaten für die Energietechnik.
Habilitation, TU Dresden, Fakultät Maschinenwesen (1990)

[14] VDI Richtlinie 4670
Thermodynamische Stoffwerte von feuchter Luft und Verbrennungsgasen.
VDI-Handbuch Energietechnik (2000)

[15] Lemmon, E. W.; Jacobsen, R. T; Penoncello, S. G.; Friend, D. G.:
Thermodynamic Properties of Air and Mixtures of Nitrogen, Argon and Oxygen from
60 to 2000 K at Pressures to 2000 MPa.
Journal of Physical Chemical Reference Data 29 (2000) Nr. 3, S. 331-385

[16] Baehr, H.D.; Tillner- Roth,R.:
Thermodynamische Eigenschaften umweltverträglicher Kältemittel, Zustandsgleichungen und Tafeln für Ammoniak, R22, R134a, R152a und R 123.
Springer-Verlag, Berlin Heidelberg (1995)

[17] Fenghour, A.; Wakeham, W. A.; Vesovic, V.; Watson, J. T. R.; Millat, J.; Vogel, E.:
The Viskosity of Ammonia.
J. Phys. Chem. Ref. Data, 24, (1995) Nr. 5, S. 1649-1667

[18] Tufeu, R.; Ivanov, D. Y.; Garrabos, Y.; Le Neindre, B.:
Thermal Conductivity of Ammonia in a Large Temperature and Pressure Range Including the Critical Region.
Ber. Bunsenges. Phys. Chem. 88 (1984) S. 422-427

[19] Span, R.; Wagner W.:
A New Equation of State for Carbon Dioxide Covering the Fluid Region from the Triple-Point Temperature to 1100 K at Pressures up to 800 MPa.
J. Phys. Chem. Ref. Data, 25, (1996) Nr. 6, S. 1506-1596

[20] Vesovic, V.; Wakeham, W. A.; Olchowy, G. A.; Sengers, J. V.; Watson, J. T. R.; Millat, J.:
The Transport Properties of Carbon Dioxide.
J. Phys. Chem. Ref. Data, 19, (1990) Nr. 3, S. 763-808

[21] Kim D. S.; Infante Ferreira, C. A.:
A Gibbs Energy Equation for LiBr/H_2O Solutions.
Proceedings of the 6th IIR Gustav Lorentzen Conference on Natural Working Fluids, Glasgow (U.K.), 2004

[22] Kim D. S.; Infante Ferreira, C. A.:
Solar Absorption Cooling.
1st progress report for NOVEM, Oct, 2003

[23] Vollmer, D.:
Programm zur Berechnung thermodynamischer Eigenschaften des Arbeitsstoffpaares Wasser/Lithiumbromid.
Institut für Luft- und Kältetechnik Dresden, intern

[24] Hasch, S.:
Diplomarbeit: Berechnung der thermodynamischen Stoffeigenschaften von binären Stoffgemischen in Absorptions- und Desorptionsprozessen der Kältetechnik.
Hochschule Zittau/Görlitz (FH), 2005

6. Satisfied Customers

Date: 10/2011

The following companies and institutions use the property libraries

- FluidEXL^{Graphics} for Excel®
- FluidLAB for MATLAB®
- FluidMAT for Mathcad®
- FluidEES for Engineering Equation Solver® EES
- FluidDYM for Dymola® (Modelica)
- FluidVIEW for LabVIEW®:

2011

Lopez, Munguia, Spain	10/2011
University of KwaZulu-Natal, Westville, South Africa	10/2011
Voith, Heidenheim	09/2011
SpgBe Montreal, Canada	09/2011
SPG TECH, Montreuil Cedex, France	09/2011
Voith, Heidenheim-Mergelstetten	09/2011
MTU Aero Engines, Munich	08/2011
MIBRAG, Zeitz	08/2011
RWE, Essen	07/2011
Fels, Elingerode	07/2011
Weihenstephan University of Applied Sciences	07/2011, 09/2011, 10/2011
Forschungszentrum Juelich	07/2011
RWTH Aachen University	07/2011, 08/2011
INNEO Solutions, Ellwangen	06/2011
Caliqua, Basel, Switzerland	06/2011
Technical University of Freiberg	06/2011
Fichtner IT Consulting, Stuttgart	05/2011, 06/2011, 08/2011
Salzgitter Flachstahl, Salzgitter	05/2011
Helbling Beratung & Bauplanung, Zurich, Switzerland	05/2011
INEOS, Cologne	04/2011
Enseleit Consulting Engineers, Siebigerode	04/2011
Witt Consulting Engineers, Stade	03/2011

Helbling, Zurich, Switzerland	03/2011
MAN Diesel, Copenhagen, Denmark	03/2011
AGO, Kulmbach	03/2011
University of Duisburg	03/2011, 06/2011
CCP, Marburg	03/2011
BASF, Ludwigshafen	02/2011
ALSTOM Power, Baden, Switzerland	02/2011
Universität der Bundeswehr, Munich	02/2011
Calorifer, Elgg, Switzerland	01/2011
STRABAG, Vienna, Austria	01/2011
TUEV Sued, Munich	01/2011
ILK Dresden	01/2011
Technical University of Dresden	01/2011, 05/2011, 06/2011, 08/2011

2010

Umweltinstitut Neumarkt	12/2010
YIT Austria, Vienna, Austria	12/2010
MCI Innsbruck, Austria	12/2010
University of Stuttgart	12/2010
HS Cooler, Wittenburg	12/2010
Visteon, Novi Jicin, Czech Republic	12/2010
CompuWave, Brunntal	12/2010
Stadtwerke Leipzig	12/2010
MCI Innsbruck, Austria	12/2010
EVONIK Energy Services, Zwingenberg	12/2010
Caliqua, Basel, Switzerland	11/2010
Shanghai New Energy Resources Science & Technology, China	11/2010
Energieversorgung Halle	11/2010
Hochschule für Technik Stuttgart, University of Applied Sciences	11/2010
Steinmueller, Berlin	11/2010
Amberg-Weiden University of Applied Sciences	11/2010
AREVA NP, Erlangen	10/2010
MAN Diesel, Augsburg	10/2010
KRONES, Neutraubling	10/2010
Vaillant, Remscheid	10/2010

PC Ware, Leipzig	10/2010
Schubert Consulting Engineers, Weißenberg	10/2010
Fraunhofer Institut UMSICHT, Oberhausen	10/2010
Behringer Consulting Engineers, Tagmersheim	09/2010
Saacke, Bremen	09/2010
WEBASTO, Neubrandenburg	09/2010
Concordia University, Montreal, Canada	09/2010
Compañía Eléctrica de Sochagota, Bogota, Colombia	08/2010
Hannover University of Applied Sciences	08/2010
ERGION, Mannheim	07/2010
Fichtner IT Consulting, Stuttgart	07/2010
TF Design, Matieland, South Africa	07/2010
MCE, Berlin	07/2010, 12/2010
IPM, Zittau/Goerlitz University of Applied Sciences	06/2010
TUEV Sued, Dresden	06/2010
RWE IT, Essen	06/2010
Glen Dimplex, Kulmbach	05/2010, 07/2010 10/2010
Hot Rock, Karlsruhe	05/2010
Darmstadt University of Applied Sciences	05/2010
Voith, Heidenheim	04/2010
CombTec, Zittau	04/2010
University of Glasgow, Great Britain	04/2010
Universitaet der Bundeswehr, Munich	04/2010
Technical University of Hamburg-Harburg	04/2010
Vattenfall Europe, Berlin	04/2010
HUBER Consulting Engineers, Berching	04/2010
VER, Dresden	04/2010
CCP, Marburg	03/2010
Offenburg University of Applied Sciences	03/2010
Technical University of Berlin	03/2010
NIST Boulder CO, USA	03/2010
Technical University of Dresden	02/2010
Siemens Energy, Nuremberg	02/2010
Augsburg University of Applied Sciences	02/2010

ALSTOM Power, Baden, Switzerland	02/2010, 05/2010
MIT Massachusetts Institute of Technology Cambridge MA, USA	02/2010
Wieland Werke, Ulm	01/2010
Siemens Energy, Goerlitz	01/2010, 12/2010
Technical University of Freiberg	01/2010
ILK, Dresden	01/2010, 12/2010
Fischer-Uhrig Consulting Engineers, Berlin	01/2010

2009

ALSTOM Power, Baden, Schweiz	01/2009, 03/2009, 05/2009
Nordostschweizerische Kraftwerke AG, Doettingen, Switzerland	02/2009
RWE, Neurath	02/2009
Brandenburg University of Technology, Cottbus	02/2009
Hamburg University of Applied Sciences	02/2009
Kehrein, Moers	03/2009
EPP Software, Marburg	03/2009
Bernd Münstermann, Telgte	03/2009
Suedzucker, Zeitz	03/2009
CPP, Marburg	03/2009
Gelsenkirchen University of Applied Sciences	04/2009
Regensburg University of Applied Sciences	05/2009
Gatley & Associates, Atlanta, USA	05/2009
BOSCH, Stuttgart	06/2009, 07/2009
Dr. Nickolay, Consulting Engineers, Gommersheim	06/2009
Ferrostal Power, Saarlouis	06/2009
BHR Bilfinger, Essen	06/2009
Intraserv, Wiesbaden	06/2009
Lausitz University of Applied Sciences, Senftenberg	06/2009
Nuernberg University of Applied Sciences	06/2009
Technical University of Berlin	06/2009
Fraunhofer Institut UMSICHT, Oberhausen	07/2009
Bischoff, Aurich	07/2009
Fichtner IT Consulting, Stuttgart	07/2009
Techsoft, Linz, Austria	08/2009
DLR, Stuttgart	08/2009

Wienstrom, Vienna, Austria	08/2009
RWTH Aachen University	09/2009
Vattenfall, Hamburg	10/2009
AIC, Chemnitz	10/2009
Midiplan, Bietigheim-Bissingen	11/2009
Institute of Air Handling and Refrigeration ILK, Dresden	11/2009
FZD, Rossendorf	11/2009
Techgroup, Ratingen	11/2009
Robert Sack, Heidelberg	11/2009
EC, Heidelberg	11/2009
MCI, Innsbruck, Austria	12/2009
Saacke, Bremen	12/2009
ENERKO, Aldenhoven	12/2009

2008

Pink, Langenwang	01/2008
Fischer-Uhrig, Berlin	01/2008
University of Karlsruhe	01/2008
MAAG, Kuesnacht, Switzerland	02/2008
M&M Turbine Technology, Bielefeld	02/2008
Lentjes, Ratingen	03/2008
Siemens Power Generation, Goerlitz	04/2008
Evonik, Zwingenberg (general EBSILON program license)	04/2008
WEBASTO, Neubrandenburg	04/2008
CFC Solutions, Munich	04/2008
RWE IT, Essen	04/2008
Rerum Cognitio, Zwickau	04/2008, 05/2008
ARUP, Berlin	05/2008
Research Center, Karlsruhe	07/2008
AWECO, Neukirch	07/2008
Technical University of Dresden, Professorship of Building Services	07/2008
Technical University of Cottbus, Chair in Power Plant Engineering	07/2008, 10/2008
Ingersoll-Rand, Unicov, Czech Republic	08/2008
Technip Benelux BV, Zoetermeer, Netherlands	08/2008
Fennovoima Oy, Helsinki, Finland	08/2008

Fichtner Consulting & IT, Stuttgart	09/2008
PEU, Espenhain	09/2008
Poyry, Dresden	09/2008
WINGAS, Kassel	09/2008
TUEV Sued, Dresden	10/2008
Technical University of Dresden, Professorship of Thermic Energy Machines and Plants	10/2008, 11/2008
AWTEC, Zurich, Switzerland	11/2008
Siemens Power Generation, Erlangen	12/2008

2007

Audi, Ingolstadt	02/2007
ANO Abfallbehandlung Nord, Bremen	02/2007
TUEV NORD SysTec, Hamburg	02/2007
VER, Dresden	02/2007
Technical University of Dresden, Chair in Jet Propulsion Systems	02/2007
Redacom, Nidau, Switzerland	02/2007
Universität der Bundeswehr, Munich	02/2007
Maxxtec, Sinsheim	03/2007
University of Rostock, Chair in Technical Thermodynamics	03/2007
AGO, Kulmbach	03/2007
University of Stuttgart, Chair in Aviation Propulsions	03/2007
Siemens Power Generation, Duisburg	03/2007
ENTHAL Haustechnik, Rees	05/2007
AWECO, Neukirch	05/2007
ALSTOM, Rugby, Great Britain	06/2007
SAAS, Possendorf	06/2007
Grenzebach BSH, Bad Hersfeld	06/2007
Reichel Engineering, Haan	06/2007
Technical University of Cottbus, Chair in Power Plant Engineering	06/2007
Voith Paper Air Systems, Bayreuth	06/2007
Egger Holzwerkstoffe, Wismar	06/2007
Tissue Europe Technologie, Mannheim	06/2007
Dometic, Siegen	07/2007
RWTH Aachen University, Institute for Electrophysics	09/2007
National Energy Technology Laboratory, Pittsburg, USA	10/2007

Energieversorgung Halle	10/2007
AL-KO, Jettingen	10/2007
Grenzebach BSH, Bad Hersfeld	10/2007
Wiesbaden University of Applied Sciences, Department of Engineering Sciences	10/2007
Endress+Hauser Messtechnik, Hannover	11/2007
Munich University of Applied Sciences, Department of Mechanical Engineering	11/2007
Rerum Cognitio, Zwickau	12/2007
Siemens Power Generation, Erlangen	11/2007
University of Rostock, Chair in Technical Thermodynamics	11/2007, 12/2007

2006

STORA ENSO Sachsen, Eilenburg	01/2006
Technical University of Munich, Chair in Energy Systems	01/2006
NUTEC Engineering, Bisikon, Switzerland	01/2006, 04/2006
Conwel eco, Bochov, Czech Republic	01/2006
Offenburg University of Applied Sciences	01/2006
KOCH Transporttechnik, Wadgassen	01/2006
BEG Bremerhavener Entsorgungsgesellschaft	02/2006
Deggendorf University of Applied Sciences, Department of Mechanical Engineering and Mechatronics	02/2006
University of Stuttgart, Department of Thermal Fluid Flow Engines	02/2006
Technical University of Munich, Chair in Apparatus and Plant Engineering	02/2006
Energietechnik Leipzig (company license),	02/2006
Siemens Power Generation, Erlangen	02/2006, 03/2006
RWE Power, Essen	03/2006
WAETAS, Pobershau	04/2006
Siemens Power Generation, Goerlitz	04/2006
Technical University of Braunschweig, Department of Thermodynamics	04/2006
EnviCon & Plant Engineering, Nuremberg	04/2006
Brassel Engineering, Dresden	05/2006
University of Halle-Merseburg, Department of USET Merseburg incorporated society	05/2006
Technical University of Dresden, Professorship of Thermic Energy Machines and Plants	05/2006

Fichtner Consulting & IT Stuttgart (company licenses and distribution)	05/2006
Suedzucker, Ochsenfurt	06/2006
M&M Turbine Technology, Bielefeld	06/2006
Feistel Engineering, Volkach	07/2006
ThyssenKrupp Marine Systems, Kiel	07/2006
Caliqua, Basel, Switzerland (company license)	09/2006
Atlas-Stord, Rodovre, Denmark	09/2006
Konstanz University of Applied Sciences, Course of Studies Construction and Development	10/2006
Siemens Power Generation, Duisburg	10/2006
Hannover University of Applied Sciences, Department of Mechanical Engineering	10/2006
Siemens Power Generation, Berlin	11/2006
Zikesch Armaturentechnik, Essen	11/2006
Wismar University of Applied Sciences, Seafaring Department	11/2006
BASF, Schwarzheide	12/2006
Enertech Energie und Technik, Radebeul	12/2006

2005

TUEV Nord, Hannover	01/2005
J.H.K Plant Engineering and Service, Bremerhaven	01/2005
Electrowatt-EKONO, Zurich, Switzerland	01/2005
FCIT, Stuttgart	01/2005
Energietechnik Leipzig (company license)	02/2005, 04/2005, 07/2005
eta Energieberatung, Pfaffenhofen	02/2005
FZR Forschungszentrum, Rossendorf/Dresden	04/2005
University of Saarbruecken	04/2005
Technical University of Dresden Professorship of Thermic Energy Machines and Plants	04/2005
Grenzebach BSH, Bad Hersfeld	04/2005
TUEV Nord, Hamburg	04/2005
Technical University of Dresden, Waste Management	05/2005
Siemens Power Generation, Goerlitz	05/2005
Duesseldorf University of Applied Sciences, Department of Mechanical Engineering and Process Engineering	05/2005

Redacom, Nidau, Switzerland	06/2005
Dumas Verfahrenstechnik, Hofheim	06/2005
Alensys Engineering, Erkner	07/2005
Stadtwerke Leipzig	07/2005
SaarEnergie, Saarbruecken	07/2005
ALSTOM ITC, Rugby, Great Britain	08/2005
Technical University of Cottbus, Chair in Power Plant Engineering	08/2005
Vattenfall Europe, Berlin (group license)	08/2005
Technical University of Berlin	10/2005
Basel University of Applied Sciences, Department of Mechanical Engineering, Switzerland	10/2005
Midiplan, Bietigheim-Bissingen	11/2005
Technical University of Freiberg, Chair in Hydrogeology	11/2005
STORA ENSO Sachsen, Eilenburg	12/2005
Energieversorgung Halle (company license)	12/2005
KEMA IEV, Dresden	12/2005

2004

Vattenfall Europe (group license)	01/2004
TUEV Nord, Hamburg	01/2004
University of Stuttgart, Institute of Thermodynamics and Heat Engineering	02/2004
MAN B&W Diesel A/S, Copenhagen, Denmark	02/2004
Siemens AG Power Generation, Erlangen	02/2004
Ulm University of Applied Sciences	03/2004
Visteon, Kerpen	03/2004, 10/2004
Technical University of Dresden, Professorship of Thermic Energy Machines and Plants	04/2004
Rerum Cognitio, Zwickau	04/2004
University of Saarbruecken	04/2004
Grenzebach BSH, Bad Hersfeld	04/2004
SOFBID Zwingenberg (general EBSILON program license)	04/2004
EnBW Energy Solutions, Stuttgart	05/2004
HEW-Kraftwerk, Tiefstack	06/2004
h s energieanlagen, Freising	07/2004
FCIT, Stuttgart	08/2004
Physikalisch Technische Bundesanstalt (PTB), Braunschweig	08/2004
Mainova Frankfurt	08/2004

Rietschle Energieplaner, Winterthur, Switzerland	08/2004
MAN Turbo Machines, Oberhausen	09/2004
TUEV Sued, Dresden	10/2004
STEAG Kraftwerk, Herne	10/2004, 12/2004
University of Weimar	10/2004
energeticals (e-concept), Munich	11/2004
SorTech, Halle	11/2004
Enertech EUT, Radebeul (company license)	11/2004
Munich University of Applied Sciences	12/2004
STORA ENSO Sachsen, Eilenburg	12/2004
Technical University of Cottbus, Chair in Power Plant Engineering	12/2004
Freudenberg Service, Weinheim	12/2004

2003

Paper Factory, Utzenstorf, Switzerland	01/2003
MAB Plant Engineering, Vienna, Austria	01/2003
Wulff Energy Systems, Husum	01/2003
Technip Benelux BV, Zoetermeer, Netherlands	01/2003
ALSTOM Power, Baden, Switzerland	01/2003, 07/2003
VER, Dresden	02/2003
Rietschle Energieplaner, Winterthur, Switzerland	02/2003
DLR, Leupholdhausen	04/2003
Emden University of Applied Sciences, Department of Technology	05/2003
Petterssson+Ahrends, Ober-Moerlen	05/2003
SOFBID ,Zwingenberg (general EBSILON program license)	05/2003
Ingenieurbuero Ostendorf, Gummersbach	05/2003
TUEV Nord, Hamburg	06/2003
Muenstermann GmbH, Telgte-Westbevern	06/2003
University of Cali, Colombia	07/2003
Atlas-Stord, Rodovre, Denmark	08/2003
ENERKO, Aldenhoven	08/2003
STEAG RKB, Leuna	08/2003
eta Energieberatung, Pfaffenhofen	08/2003
exergie, Dresden	09/2003
AWTEC, Zurich, Switzerland	09/2003
Energie, Timelkam, Austria	09/2003

Electrowatt-EKONO, Zurich, Switzerland	09/2003
LG, Annaberg-Buchholz	10/2003
FZR Forschungszentrum, Rossendorf/Dresden	10/2003
EnviCon & Plant Engineering, Nuremberg	11/2003
Visteon, Kerpen	11/2003
VEO Vulkan Energiewirtschaft Oderbruecke, Eisenhuettenstadt	11/2003
Stadtwerke Hannover	11/2003
SaarEnergie, Saarbruecken	11/2003
Fraunhofer-Gesellschaft, Munich	12/2003
Erfurt University of Applied Sciences, Department of Supply Engineering	12/2003
SorTech, Freiburg	12/2003
Mainova, Frankfurt	12/2003
Energieversorgung Halle	12/2003

2002

Hamilton Medical AG, Rhaezuens, Switzerland	01/2002
Bochum University of Applied Sciences, Department of Thermo- and Fluid Dynamics	01/2002
SAAS, Possendorf/Dresden	02/2002
Siemens, Karlsruhe (general license for the WinIS information system)	02/2002
FZR Forschungszentrum, Rossendorf/Dresden	03/2002
CompAir, Simmern	03/2002
GKS Gemeinschaftskraftwerk, Schweinfurt	04/2002
ALSTOM Power Baden, Switzerland (group licenses)	05/2002
InfraServ, Gendorf	05/2002
SoftSolutions, Muehlhausen (company license)	05/2002
DREWAG, Dresden (company license)	05/2002
SOFBID, Zwingenberg (general EBSILON program license)	06/2002
Kleemann Engineering, Dresden	06/2002
Caliqua, Basel, Switzerland (company license)	07/2002
PCK Raffinerie, Schwedt (group license)	07/2002
Fischer-Uhrig Engineering, Berlin	08/2002
Fichtner Consulting & IT, Stuttgart (company licenses and distribution)	08/2002
Stadtwerke Duisburg	08/2002

Stadtwerke Hannover	09/2002
Siemens Power Generation, Goerlitz	10/2002
Energieversorgung Halle (company license)	10/2002
Bayer, Leverkusen	11/2002
Dillinger Huette, Dillingen	11/2002
G.U.N.T. Geraetebau, Barsbuettel (general license and training test benches)	12/2002
VEAG, Berlin (group license)	12/2002

2001

ALSTOM Power, Baden, Switzerland	01/2001, 06/2001, 12/2001
KW2 B. V., Amersfoot, Netherlands	01/2001, 11/2001
Eco Design, Saitamaken, Japan	01/2001
M&M Turbine Technology, Bielefeld	01/2001, 09/2001
MVV Energie, Mannheim	02/2001
Technical University of Dresden, Department of Power Machinery and Plants	02/2001
PREUSSAG NOELL, Wuerzburg	03/2001
Fichtner Consulting & IT Stuttgart (company licenses and distribution)	04/2001
Muenstermann GmbH, Telgte-Westbevern	05/2001
SaarEnergie, Saarbruecken	05/2001
Siemens, Karlsruhe (general license for the WinIS information system)	08/2001
Neusiedler AG, Ulmerfeld, Austria	09/2001
h s energieanlagen, Freising	09/2001
Electrowatt-EKONO, Zurich, Switzerland	09/2001
IPM Zittau/Goerlitz University of Applied Sciences (general license)	10/2001
eta Energieberatung, Pfaffenhofen	11/2001
ALSTOM Power Baden, Switzerland	12/2001
VEAG, Berlin (group license)	12/2001

2000

SOFBID, Zwingenberg (general EBSILON program license)	01/2000
AG KKK - PGW Turbo, Leipzig	01/2000
PREUSSAG NOELL, Wuerzburg	01/2000
M&M Turbine Technology, Bielefeld	01/2000

IBR Engineering Reis, Nittendorf-Undorf	02/2000
GK, Hannover	03/2000
KRUPP-UHDE, Dortmund (company license)	03/2000
UMAG W. UDE, Husum	03/2000
VEAG, Berlin (group license)	03/2000
Thinius Engineering, Erkrath	04/2000
SaarEnergie, Saarbruecken	05/2000, 08/2000
DVO Data Processing Service, Oberhausen	05/2000
RWTH Aachen University	06/2000
VAUP Process Automation, Landau	08/2000
Knuerr-Lommatec, Lommatzsch	09/2000
AVACON, Helmstedt	10/2000
Compania Electrica, Bogota, Colombia	10/2000
G.U.N.T. Geraetebau, Barsbuettel (general license for training test benches)	11/2000
Steinhaus Informationssysteme, Datteln (general license for process data software)	12/2000

1999

Bayernwerk, Munich	01/1999
DREWAG, Dresden (company license)	02/1999
KEMA IEV, Dresden	03/1999
Regensburg University of Applied Sciences	04/1999
Fichtner Consulting & IT, Stuttgart (company licenses and distribution)	07/1999
Technical University of Cottbus, Chair in Power Plant Engineering	07/1999
Technical University of Graz, Department of Thermal Engineering, Austria	11/1999
Ostendorf Engineering, Gummersbach	12/1999

1998

Technical University of Cottbus, Chair in Power Plant Engineering	05/1998
Fichtner Consulting & IT (CADIS information systems) Stuttgart (general KPRO program license)	05/1998
M&M Turbine Technology Bielefeld	06/1998
B+H Software Engineering Stuttgart	08/1998
Alfa Engineering, Switzerland	09/1998
VEAG Berlin (group license)	09/1998
NUTEC Engineering, Bisikon, Switzerland	10/1998

SCA Hygiene Products, Munich	10/1998
RWE Energie, Neurath	10/1998
Wilhelmshaven University of Applied Sciences	10/1998
BASF, Ludwigshafen (group license)	11/1998
Energieversorgung, Offenbach	11/1998

1997

Gerb, Dresden	06/1997
Siemens Power Generation, Goerlitz	07/1997